A Two-component Model of Texture for Analysis and Synthesis

Tao-I Hsu†and R Wilson
University of Warwick
Department of Computer Science
Coventry CV4 7AL
England.
†Chung Cheng Institute of Technology
Taiwan
R.O.C.

December 11, 1995

Abstract

A model of natural texture based on a structural component which uses affine coordinate transformations and a stochastic residual component is presented. It is argued that the selection of an appropriate analysis scale can be formulated in terms of a trade-off between the rate at which parameters are generated and the distortion resulting from the approximation by the structural component. An efficient algorithm for identifying the parameters of the structural model is described and its utility demonstrated on a number of synthetic and natural textures.

1 Introduction

The analysis and synthesis of natural textures is one of the most intriguing and difficult problems in image processing, which has received much attention over the years [2, 20, 26, 21, 16, 10, 34, 35, 5, 4, 17]. Apart from their significance in applications ranging from remote sensing to computer graphics, the analysis and synthesis of texture have remained challenging problems because of the combination of structure and variation which many textures possess: simple structural or statistical models fail to capture the subtlety of this combination in many cases of practical interest. Indeed, while 'texture' has its roots in the production of textiles, implying a more or less periodic structure, there is no universally accepted definition of texture which may be applied to all the relevant imagery - even an array of letters may be regarded as a texture, whose discrimination by human subjects appears to be conditional on typical textural features [25].

The observation that a given neighbourhood of an image may both be seen as containing some definite structure, such as a letter or edge feature, while also being seen as a part of a texture, shows that texture is a scale-dependent phenomenon. In textures having a definite periodicity, the scale is obviously determined by the period, but in general it may be harder to identify a scale unambiguously. Indeed, the so-called 'fractional Brownian motion' models, which are widley used in image synthesis, display structure over a wide range of wavelengths, in a way which is a statistical analogue of fractal patterns [31, 27]. Correspondingly, there has been an increasing use of multiresolution modelling and analysis methods over the last decade or so (eg. [28, 24, 35, 32, 4, 13]), a trend which looks set to continue under the impetus of developments in wavelet theory. During the same period, Markov Random Field (MRF) models, based on both causal and non-causal neighbourhoods, have found widespread use [20, 26, 10, 18, 12, 9, 14]. In the simplest case, linear models based on a causal or semi-causal neighbourhood, the comparative simplicity of model identification is somewhat offset by the limited range of textures which can be produced, while more complex, nonlinear and non-causal models require large amounts of processing, but can model a significantly wider range of textures. Non-parametric methods show a similar trade-off between computational demands and realism: second order methods, such as spectrum estimation [35, 5] are not capable of discriminating textures having quite different visual characteristics, while higher order methods are expensive. It would be a useful step forward to develop a model with more power than traditional techniques, but which did not require massive computational resources. This seems to demand a marriage between structural and statistical techniques.

Any attempt to combine structural and statistical elements ought to start from an exploration of the sources of structure in texture. One elementary observation is that natural textures are generally caused by the surface properties of objects, which are three dimensional. It follows that a good structural model ought to be able to approximate the effects of $3-D$ motions, at least locally. This suggests a $2-D$ affine model of motion [17], in which the *local coordinates* in two separate regions of the image are related by an affine transformation: $\vec{\xi}_2 = T_{21}\vec{\xi}_1$. The idea of using affine transformations to approximate the $3-D$ geometric relationship between patches of texture underlies the work of Gårding on surface orientation estimation [17] and is similar to the warping model of Volet [34]. It also shares a number of features with the fractal methods developed by Barnsley and his colleagues [3]. Although a description of texture in terms of a notional underlying surface geometry is appealing, it will be inadequate to deal with all of the variability encountered in natural textures: a statistical element is still required to deal with the random elements, whether they are caused by imperfections in the imaging process or on the surface imaged.

The model of texture proposed in the present work combines a structural element based on $2-D$ affine relationships between texture patches with a stochastic element designed to account for the unpredictable variations in the texture in a manner analogous to a Wold decomposition [15]. The next section contains an account of the new model. Examples are given of textures generated using the model and the issue of scale is addressed. A description is then given of an 'analysis-by-synthesis' algorithm, which has been applied succesfully to the modelling of a number of textures taken from Brodatz's book of textures [6]. The paper is concluded with a discussion of the implications of the work.

2 Self-similarity in Textures

The simple notion of texture as a repetition of some pattern element, or *texel*, while inadequate for many interesting textures, captures one important aspect of texture: textures are *self-similar*. Thus if $x(\vec{\xi})$ denotes the texture as a function of the $2-D$ image coordinate $\vec{\xi}$, then periodicity amounts to the relationship

$$x(\vec{\xi}) = x(\vec{\xi} - m\vec{\alpha} - n\vec{\beta}), \quad m, n \in Z \tag{1}$$

while fractal self-similarity can be defined through

$$x(\vec{\xi}) = x(\delta^m \vec{\xi}), \quad m \in Z \tag{2}$$

which is the analogue of (1) in scale. The significant feature of both (2) and (1) is that it specifies a method of synthesis in each case: take a 'patch' of texture of an appropriate size and add to it patches whose coordinates have been transformed by the actions of the group elements. Thus if $x(\vec{\xi})$ denotes a patch whose support is the unit square

$$x(\vec{\xi}) = 0, \quad \vec{\xi} \notin [0,1) \times [0,1) \tag{3}$$

a texture which is self-similar under translations can be produced by

$$s(\vec{\xi}) = \sum_{m,n} x(\vec{\xi} - m\vec{\alpha} - n\vec{\beta}) \tag{4}$$

While a similar device can be used to synthesise a fractal texture from the patch, there is clearly no way of combining the two to give a texture which is self-similar with respect to both translations and dilations. It is, however, possible to generate textures in which a given patch is arrayed on a regular cartesian grid, having first been transformed in some way:

$$s(\vec{\xi}) = \sum_{m,n} w(\vec{\xi} - m\vec{\alpha} - n\vec{\beta}) x(\boldsymbol{T}_{mn}(\vec{\xi} - m\vec{\alpha} - n\vec{\beta})) \tag{5}$$

where $\boldsymbol{T}_{mn}$ is chosen from a suitable group of coordinate transforms $\boldsymbol{T}_{mn} : R^2 \rightarrow R^2$, such as the $2-D$ affine group and $w(\vec{\xi})$ is a *window* function. With the right relationship between the window, the periods $\vec{\alpha}, \vec{\beta}$ and the translation component of the transformations $\boldsymbol{T}_{mn}$, it is possible to 'stitch' together a seamless texture whose period is *independent* of the sampling intervals $\vec{\alpha}, \vec{\beta}$. An example of such a synthesis is shown in figure 1. This example is based on randomly chosen affine transformations operating on a patch size of 32×32 pixels taken from the reptile skin of figure 6, on a cartesian synthesis lattice with a 16×16 pixel cell, using a $cos()^2$ window ensuring proper interpolation: there is no obvious sign of either the windowing or the synthesis lattice in this result. Clearly, this is a useful extension of the concept of self-similarity, which can produce realistic textures from a relatively small patch of an original texture image.

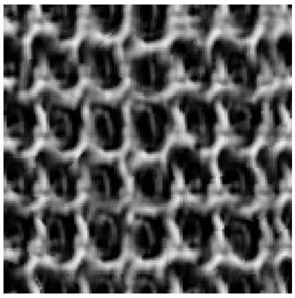


Figure 1: Texture synthesis based solely on coordinate transformations.

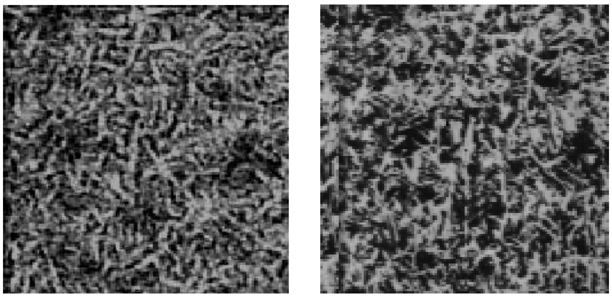
As a general texture model, however, there is something missing from this formulation - the above example is taken from a texture with a relatively high degree of structure. What happens if the original texture is less structured? A glance at figure 8 shows the problem: random textures acquire *too much* structure when modelled in this way. This is hardly surprising, since there is generally significantly more information per pixel in a random texture than the small quantity contained in the 6 affine parameters per lattice point (ie. 0.023 parameters/pixel). An obvious way to increase the range of textures significantly is to add to the texture defined in (5) a stochastic component, giving

$$s(\vec{\xi}) = \sum_{m,n} w(\vec{\xi} - m\vec{\alpha} - n\vec{\beta}) x(\boldsymbol{T}_{mn}(\vec{\xi} - m\vec{\alpha} - n\vec{\beta})) + e(\vec{\xi}) \tag{6}$$

where $e(\vec{\xi})$ is a sample from a suitable random texture process, such as an MRF. In many practical applications, a sample of natural texture may be given as $s(\vec{\xi})$ and the noise process will then be defined in terms of the residual error

$$e(\vec{\xi}) = s(\vec{\xi}) - s^a(\vec{\xi}) \tag{7}$$

where $s^a(\vec{\xi})$ is the approximation of $s(\vec{\xi})$ based on the affine model of (5). As an illustration of the idea, figure 2 shows a 128×128 pixel sample from figure 8 synthe-



(a) (b)

Figure 2: (a) Texture synthesis based on a combination of affine coordinate transformations and a stochastic residual and (b) the corresponding part of the original texture image.

sised at a block size of 16×16 pixels, to which has been added stationary noise taken from a normal process whose energy spectrum is given by that of the residual errors, $e(\vec{\xi})$. Although not a perfect match for the grass texture, it is clearly a much better approximation visually than the simple affine one of figure 8(c). In effect, the texture is being modelled as the sum of a 'deterministic component', from the affine model, and a stochastic one. Such techniques have been used in audio signal analysis and synthesis for some time [33] and have recently been applied to texture modelling [15]. It is important not to push the analogy too far, however: the structural component defined by the affine model is not deterministic - the randomness is in the model parameters, rather than the model input.

A fundamental issue in this form of texture modelling is that of scale: is there an optimum scale at which to decompose the texture into its two components and how could we estimate that scale for a given texture? One way forward is to regard the problem as a form of *rate-distortion* problem: a given lattice spacing defines an information rate at which affine parameters are generated, since there is a fixed number of parameters per lattice point, each of which will require a certain resolution in bits/parameter to achieve an acceptable approximation, in terms of mean squared error (MSE) over some suitably defined texture class. It is assumed that the stochastic component - the residual - which cannot improve the approximation, in terms of squared error, achieves its effect in a manner analogous to dithering in quantization, by decorrelating the error and the signal.

As a coding problem, the analysis and synthesis of a sample of texture using the affine model bears some resemblance both to vector quantization [19] and to iterated function data compression [22]. In both of these approaches, a set of vectors is used to approximate a given patch of an image; in the former case, the set is normally selected to minimise MSE over the source, for a given cardinality, while in the latter case, the set of vectors is gleaned from the image itself and is augmented by allowing various simple transformations to be applied in the approximation process. In the present case, this is carried to an extreme: the approximations are *all* derived by transforming a single prototype. The rate-distortion problem is then to find a set of transformations, among the allowable set, such that the average information rate

$$R(D) = -E \log[p(\boldsymbol{T})] \tag{8}$$

is minimised, subject to the MSE satisfying the bound

$$d = \sum_i E(s(\vec{\xi}_i) - \boldsymbol{T} v(\vec{\xi}_i))^2 \leq D \tag{9}$$

where $v(\vec{\xi})$ is the prototype, $p(.)$ is the probability density over the transformations, induced by selecting for each texture sample the transformation minimising the MSE:

$$\sum_i (s(\vec{\xi}_i) - \boldsymbol{T} v(\vec{\xi}_i))^2 = \min_{\boldsymbol{W}} \sum_i (s(\vec{\xi}_i) - \boldsymbol{W} v(\vec{\xi}_i))^2 \tag{10}$$

The expectations are taken with respect to the source texture density in both (8) and (9). Although the rate and distortion depend on quantization of parameters as well as scale, it is scale that is the most interesting, from the point of view of the model - parameter quantization is quite straightforward. In practice, the transformations used consist of the affine transformation, magnitude scaling and addition of 4 lowpass samples per lattice point, making a total of 11 parameters per lattice point. These additions also guarantee that with an arbitrary prototype, the average distortion can be made arbitrarily low if the appropriate lattice spacing is used. In the limit, there is one lattice point/pixel and perfect reconstruction, but as the spacing increases, the error will increase, however the prototype pattern is chosen. For a given bound D in (9), there will be a maximum lattice spacing for which the bound is achieved on average. This bound relates not to any periodicity of the texture, but rather to the variability in its structure, as the experimental results will show. Thus scale in this model becomes a *relative* concept, rather than a natural texture parameter: for a given distortion D, there is a lattice spacing such that a given prototype can approximate a given class of textures with an average MSE $d \leq D$. The spacing necessary to achieve a given level of distortion does, however, depend strongly on the choice of prototype: using as a prototype a sample from the same texture will generally give significantly better results than a randomly chosen one. All of the experimental results have used prototypes selected from the texture being modelled.

3 Texture Analysis by Synthesis

Identification of the texture model requires estimates of both the structural parameters, which vary from one lattice point to another, and the stochastic model parameters, which do not. Because the latter problem has received a considerable amount of attention in the literature over the years [1, 8, 9, 12, 14, 23, 26, 29, 20], in the sequel the emphasis is on identifying the structural parameters from a given sample of texture.

The key to using an affine model for the structural component of a texture is that the affine transformations are sufficiently rich to represent the nominal motion between two texture patches. As an example, consider the case illustrated in figure 3, in which 2 regions are selected from the reptile skin image, with one of the local regions looking more or less a warped version of the other, which will be approximated by an affine coordinate map. The first problem is then to identify the best fitting affine transform parameters, given the two texture patches. ie.

$$\begin{aligned} \vec{\xi}_2 &= \boldsymbol{T}_{12}\vec{\xi}_1 \\ &= \boldsymbol{A}_{12}\vec{\xi}_1 + \vec{\gamma}_{12} \end{aligned} \tag{11}$$

where $\vec{\xi}_1$ is the coordinate in the first block, $\vec{\xi}_2$ is the corresponding coordinate in the second, and $\boldsymbol{A}_{12}$ is the linear transform from block 1 to block 2, $\vec{\gamma}_{12}$ is the relative spatial translation between them. In principle, the affine transformation $\boldsymbol{T}_{12}$ should be selected to minimise the squared error between target and transformed prototype, as in (10), but in practice the implied search represents a massive computation - equivalent to a $6-D$ correlation calculation. By selecting an appropriate image

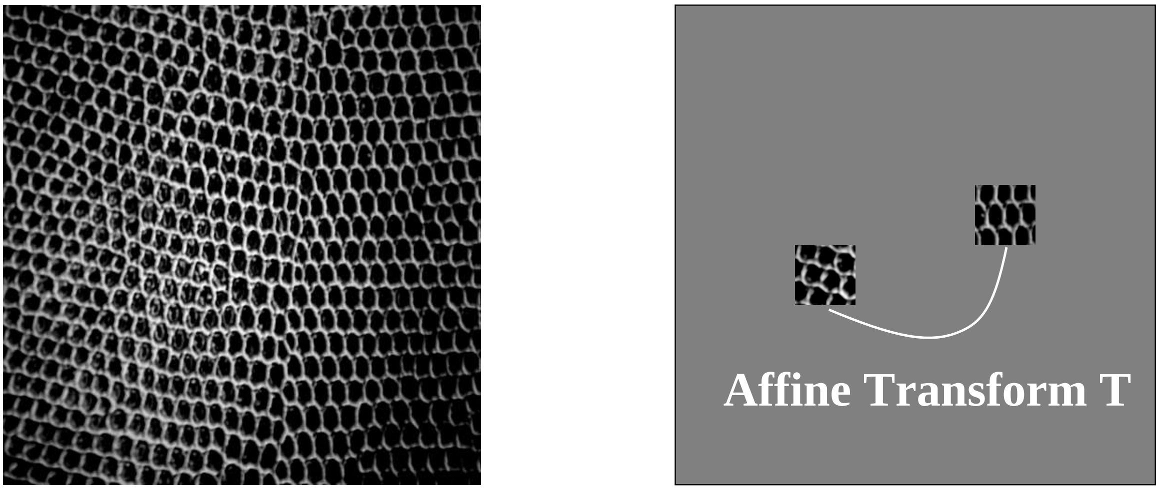


Figure 3: Texture warping is modelled via affine transform.

representation, this load can be greatly reduced without significantly affecting the quality of the result, however.

The technique chosen is based on the local Fourier spectra of the prototype and texture element of interest, for the obvious reason that the Fourier transform diagonalises translations, in effect allowing separation of the linear and translational components of $\boldsymbol{T}_{12}$. Bearing in mind the need for a suitable scale for the texture patches, a multiresolution windowed FT, the MFT, is used as the representation. Recall that a given level l of the MFT may be described as a set of estimated local spectra and is defined as [7] [11]

$$\hat{s}(\vec{\xi}_i(l), \vec{\omega}_j(l), l) = \sum_k w_l(\vec{\xi}_i(l) - \vec{\xi}_k) s(\vec{\xi}_k) \exp[-\jmath \vec{\omega}_j(l).\vec{\xi}_k] \tag{12}$$

where $\jmath^2 = -1$ and, for notational simplicity, the spatial and scale coordinates will be suppressed except where required in the sequel.

An MFT block at a given level l representing a local Fourier spectrum is first selected as the prototype at that scale. The affine transform of the prototype to each of the remaining MFT blocks is identified. On the basis of the identified transform, the original texture image can be resynthesised. As in Calway's work [7], the dominance of low frequencies is countered by a pre-emphasis step. In the present work, 4 lowpass coefficients per block are used to represent low frequencies; these are removed from the image before analysis and added after synthesis.

The implementation consists of three main components:

1. Determination of the linear coordinate transformation $\boldsymbol{A}_{12}$ which gives a best fit between the magnitude spectra.

2. Linear transformation of the prototype spectrum to give an approximation to that of the test element.

3. Correlation of the test element and transformed prototype by multiplication of the respective spectra and inverse Fourier transformation to identify the spatial translation.

Figure 4 illustrates the main components of the algorithm. For two given local spectrum estimates $\hat{s}^{(1)}(\vec{\omega}_j)$ and $\hat{s}^{(2)}(\vec{\omega}_j)$, corresponding to the two patches, $s^{(1)}(\vec{\xi}_i)$ and $s^{(2)}(\vec{\xi}_i)$, a pair of centroids is selected to represent the spectrum; from the centroid pair, a linear transform $\boldsymbol{A}$ is identified. Based on the linear transform $\boldsymbol{A}$, the prototype spectrum $\hat{s}^{(2)}(\vec{\omega}_j)$ is transformed to $\hat{s}^{(2)'}(\vec{\omega}_j)$. At this stage, the magnitude spectrum $\hat{s}^{(2)'}(\vec{\omega}_j)$ is an approximation of the test spectrum $\hat{s}^{(1)}(\vec{\omega}_j)$ except for the translation, which appears in the phase. The two spectra are multiplied and the inverse WFT applied, effectively performing a spatial correlation.

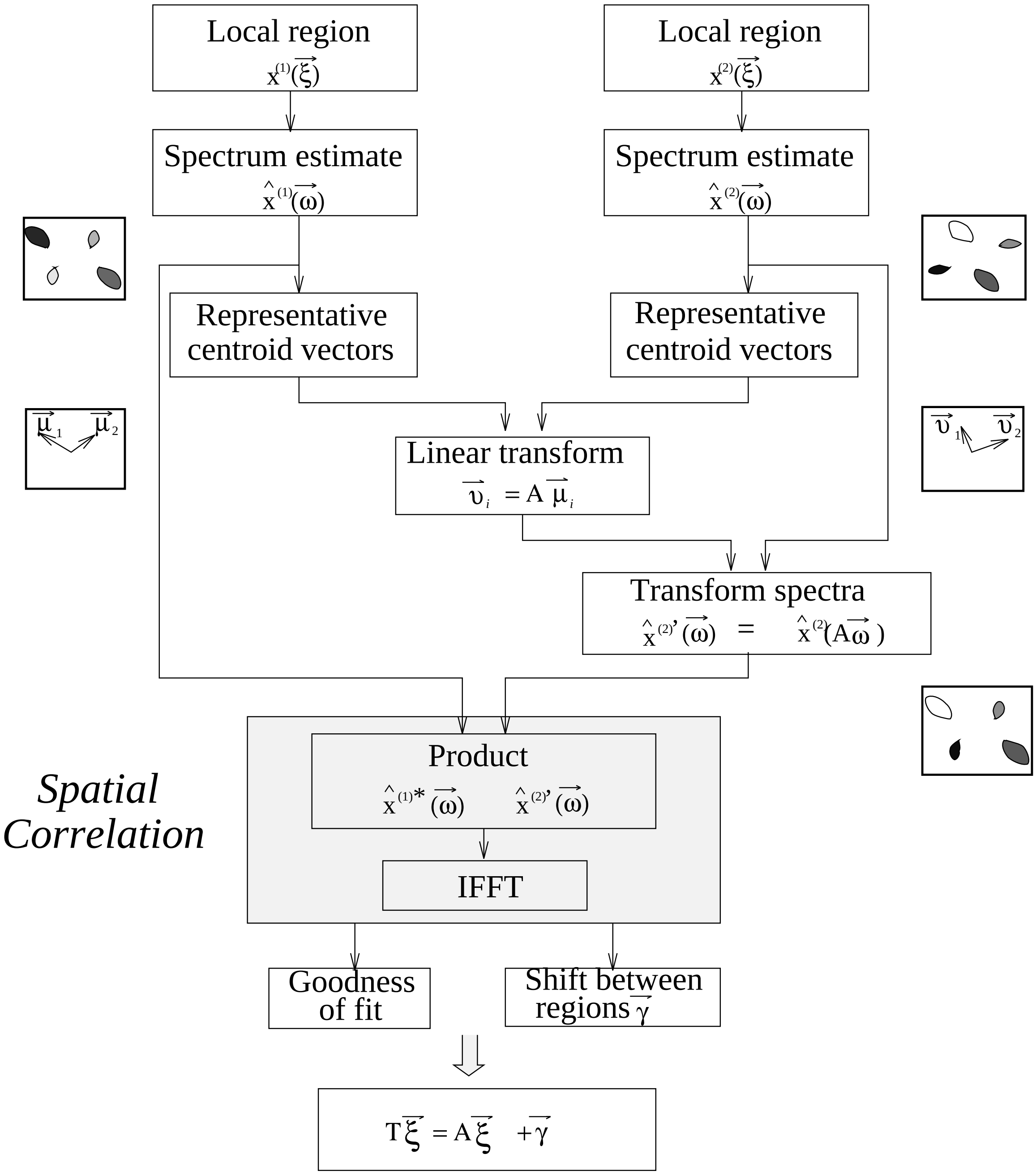


Figure 4: Outline of algorithm.

3.1 Identification of the Linear Transformation

To reduce computational complexity further, identification of the linear part of the transformation is based on a simple shape criterion: a pair of representative centroid vectors is extracted from a given local magnitude spectrum as the representation of the spectrum. These are chosen to minimise the variance of the spectral density within the corresponding segments of the frequency plane, as illustrated in figure 5:

$$\sum\nolimits^2(\theta_1,\theta_2) = \sigma_1^2(\theta_1,\theta_2) + \sigma_2^2(\theta_1,\theta_2) \tag{13}$$

is a minimum, where

$$\sigma_i^2(\theta_1,\theta_2) = \sum_{\vec{\omega}_j \in \Lambda_i(\theta_1,\theta_2)} |\hat{s}(\vec{\omega}_j)| \ \| \vec{\omega}_j - \vec{\mu}_i(\theta_1,\theta_2) \|^2 \qquad i = 1,2 \tag{14}$$

$\vec{\mu}_i(\theta_1,\theta_2)$ is the centroid

$$\vec{\mu}_i(\theta_1,\theta_2) = \frac{1}{M_i(\theta_1,\theta_2)} \sum_{\vec{\omega}_j \in \Lambda_i(\theta_1,\theta_2)} |\hat{s}(\vec{\omega}_j)|\, \vec{\omega}_j \tag{15}$$

and $M_i(\theta_1,\theta_2)$ is the total mass in the ith segment

$$M_i(\theta_1,\theta_2) = \sum_{\vec{\omega}_j \in \Lambda_i(\theta_1,\theta_2)} |\hat{s}(\vec{\omega}_j)| \tag{16}$$

$\Lambda_1(\theta_1,\theta_2)$ and $\Lambda_2(\theta_1,\theta_2)$ are the sets of coordinates in each segment of the half-plane at an angle θ_1 and divided at an angle θ_2, ie

$$\Lambda_1(\theta_1,\theta_2) = \Lambda(\theta_1) - \Lambda(\theta_1+\theta_2) \cap \Lambda(\theta_1) \tag{17}$$

$$\Lambda_2(\theta_1,\theta_2) = \Lambda(\theta_1) \cap \Lambda(\theta_1+\theta_2) \tag{18}$$

where $\Lambda(\theta)$ denotes the set of half-plane coordinates at an angle θ, i.e.

$$\omega_{j2}\cos\theta - \omega_{j1}\sin\theta \ \geq \ 0 \qquad \forall \quad \vec{\omega}_j \ \in \ \Lambda(\theta). \tag{19}$$

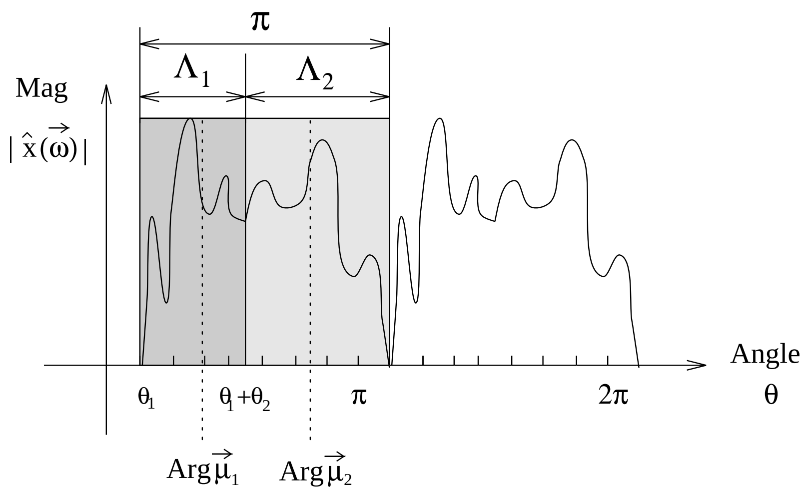


Figure 5: Representative centroid vectors; Λ_1 is coordinates between $(\theta_1, \theta_1+\theta_2)$, Λ_2 is coordinates between $(\theta_1+\theta_2, \theta_1+\pi)$, $\vec{\mu_1}$ is centroid in Λ_1 and $\vec{\mu_2}$ is centroid in Λ_2.

Identification of the linear transformation then requires solution of the equations

$$(-1)^i \vec{\mu}_j^{(1)} = \boldsymbol{A}_{12}\vec{\mu}_k^{(2)}, \quad i,j,k = 1,2 \tag{20}$$

for each of the eight distinct pairings of i, j, k in (20). Note that the term $(-1)^i$ results from the Hermitian symmetry of the FT. If the eight matrices in (20) are labelled $\boldsymbol{A}_{12}^1 - \boldsymbol{A}_{12}^8$, the prototype spectrum $\hat{s}^{(2)}(\vec{\omega}_j)$ is transformed according to

$$\hat{s}_i^{(1)}(\vec{\omega}_j) = \hat{s}^{(2)}(\boldsymbol{A}_{12}^i\vec{\omega}_j) \qquad 1 \le i \le 8 \tag{21}$$

where a given coefficient $\hat{s}_i^{(1)}(\vec{\omega}_j)$ is calculated from the discrete coefficients $\hat{s}^{(2)}(\vec{\omega}_j)$ using bilinear interpolation from its four nearest neighbours. It was shown by Calway [7] that this interpolation is adequate for estimating MFT coefficients at arbitrary coordinates $\vec{\omega}$, given the sampling scheme used in this work.

3.2 Identification of the Translation

Identification of the translation between the two patches uses the cross-correlation between them, which gives a least-squares estimate of the shift. In the present case this involves a normalised cross-correlation, which is performed in the frequency domain by applying the energy theorem [30]

$$\rho_i(\vec{\xi}_j) = \frac{\mathcal{F}^{-1}\{\hat{s}_i^{(1)}(\vec{\omega}_k)\hat{s}^{(1)*}(\vec{\omega}_k)\}}{\sum_k |\hat{s}_i^{(1)}(\vec{\omega}_k)|\ |\hat{s}^{(1)}(\vec{\omega}_k)|} \tag{22}$$

where * is the complex conjugate, $\mathcal{F}^{-1}$ denotes the inverse discrete Fourier transform (DFT), and $\vec{\xi}_j$ is the spatial coordinate. Note that normalisation is needed here to compare the quality of the 8 syntheses: bilinear interpolation does not preserve energy. The largest correlation peak, $\max_{i,j} \rho_i(\vec{\xi}_j)$, then determines the selected transformation.

3.3 Synthesis

After cross-correlation, image synthesis is accomplished by the following stages:

1. *Phase shift*: the phase shift information between the test spectrum and the best transformed spectrum is provided by the correlation calculation. This phase shift corresponds to the spatial displacement between the two signals. To correct the position of the transformed spectra, the phase term is included prior to the inverse transform:

$$\tilde{s}_k^{(1)}(\vec{\omega}_j) = \hat{s}_k^{(1)}(\vec{\omega}_j)e^{-\jmath\vec{\omega}_j\cdot\vec{\xi}_l} \tag{23}$$

where $\vec{\xi}_l$ is the position of the correlation peak.

2. *Synthesis of the texture element*: after phase correction (23), the synthesised texture element is given by

$$s_k^{(1)}(\vec{\xi}_j) = \mathcal{F}^{-1}\{\tilde{s}_k^{(1)}(\vec{\omega}_l)\} \tag{24}$$

3. *Energy normalisation*: energy normalisation is used to keep the energy of the synthesised block the same as that of the target block, ie.

$$x_k^{(1)}(\vec{\xi}_j) = \alpha_k s_k^{(1)}(\vec{\xi}_j) \tag{25}$$

where

$$\alpha_k^2 = \frac{\sum_j |\hat{s}^{(1)}(\vec{\omega}_j)|^2}{\sum_j |\hat{s}_k^{(1)}(\vec{\omega}_j)|^2} \tag{26}$$

4. *Windowing*: advantage is taken of the 50% overlap between adjacent windows in the MFT to interpolate between synthesised blocks, using a synthesis window which ensures that the overall combination of analysis and synthesis windows has a $\cos^2(.)$ response.

5. *De-emphasis*: the final step in reconstruction is addition of the lowpass component.

4 Experimental Results

Analysis and synthesis experiments based on various natural textures have been performed, in which a given texture was generated from an arbitrarily chosen prototype block at various scales. Using the appropriate local spectra, each texture block was synthesised from the prototype block using the procedure detailed above and the blocks combined to give the final synthesis. The test images represent three different types of natural texture: reptile skin, burlap and grass from Brodatz's album [6]. Results of experiments for these test images at different scales are shown in figures 6,7, and 8 respectively.

Figures 6(a) (b) (c) and (d) depict a reptile skin image size of 512×512 and its synthesis at three different levels. For all of these scales, the regularity and the warping of the texture element in the reptile skin image are reproduced satisfactorily. In figure 6(d), there are some artifacts at level 4 due to the scale of the prototype block being significantly larger than the variations in the texture. Alignment of the structure between the synthesised texture blocks is missed. Consequently, the variation of the synthesis texture at level 4 is of too low a bandwidth. On the other hand, the result at level 5 in figure 6(c) is satisfactory in terms of the SNR listed in Table 1. Figure 6(b) shows the result at the level 6, which, while of good quality visually is correspondingly more expensive in terms of information rate.

Figure 7 depicts a burlap image size of 256×256 and its synthesis at three different scales. The synthesis results demonstrate that whatever the size of the texture element, this technique still can perform satisfactorily, provided an appropriate scale is selected. In particular, the dislocation in the middle of burlap image is resynthesised. The effect of the prototype scale is again shown in figure 7(d), where the regularity has been increased so that the dislocation becomes less pronounced.

Figure 8 shows a grass image of 512×512 pixels and its synthesis at three different levels. The synthesised grass image is not as good as that of reptile and burlap in terms of the signal to noise ratio. This is because the randomness of the grass texture significantly affects the accuracy of the synthesis. Nevertheless, the result is

MFT Scale Index	Parameters Per pixel	SNR(dB)		
		Reptile	Burlap	Grass
7	0.68774	8.64229	4.04274	4.06653
6	0.17285	7.11266	3.72840	2.07278
5	0.04687	4.76636	1.79312	-0.42035
4	0.02636	2.98247	-0.21225	-1.64412

Table 1: SNR of original to synthesised texture for various scales.

reasonable, as shown in figure 8(b), which is synthesised at MFT level 7.

Table 1 lists the number of parameters used per pixel at each scale and the corresponding SNR of the synthesised texture in dB. Because the prototype is a patch taken from the same texture, the cost of this, in pixels/pixel, has been added to the rate figure based on the parameters. Note that the use of smooth, overlapping windows has prevented blocking artifacts. Moreover, the scale at which a given SNR is achieved can be seen from Table 1 to depend on the regularity of the texture, as might be expected. In figure 9, the normalised mean-square error between an original image and its synthesis is plotted as a function of the number of parameters per pixel used to synthesise it. For a normalised mean-square error of 3dB, the scale index for reptile skin is 5, for burlap 6 and for grass 7. Figure 10 overlays the block size on the corresponding images. The scale at which an acceptable result is achieved obviously depends on the definition of 'acceptable': some applications will require a more accurate reconstruction than others. Nonetheless, for a given criterion of acceptability, the scale at which an acceptable result is achieved corresponds directly to the variability of the texture. To take this argument further, it would be necessary to study the quantisation of the parameters, which goes beyond the scope of the present work.

5 Conclusion

This work has shown that a model of texture which combines structural and statistical modelling, through the use of $2-D$ affine transformations, leads to a texture description which is both general and capable of producing natural textures for reasonable cost in terms of data storage and computation. As such, it has the potential for application in many areas of image analysis and generation. The work presented has shown that the structural component of the texture can be estimated effectively and robustly from samples of diverse natural textures. A spectral method for identifying the stochastic component, based on spectral modelling of residuals, was also illustrated. Although it may be possible to achieve a similar degree of structure in some cases using MRF approaches, it is by no means clear that this can be accomplished without expending large computational resources on model identification. Moreover, the technique for structure identification can readily be applied to extraction of $3-D$ structure estimates from images of scenes containing surface textures. Overall, it seems fair to claim that it represents a useful addition to the array of methods currently available for texture analysis and generation.

The authors recognise that the approach described here has its limitations, however. More work is required on the interaction between the structural and stochastic components, for example. It may be the case in some applications that the residual spectra should be 'renormalised' using the structural model before averaging, to get a better fit to the local error. This could be accomplished with minimal extra computation or data storage. Although a criterion for selecting the scale, based on a rate-distortion trade-off, has been presented, more work seems required if an automated scale selection is to be achieved, especially if the scale is to be allowed to vary across the image. This is also related to the issue of selection of prototypes: obviously the quality of the synthesis at a given scale depends upon the choice of prototype, but the exact realtionship is not a simple one. Finally, to be of use in many applications, the method must be combined with a robust segmentation technique. All of these issues are currently under investigation.

References

[1] C. O. Acuna. Texture Modeling Using Gibbs Distributions. *CVGIP: Graphical Models and Image Processing*, 54:210–222, 1992.

[2] N. Ahuja. Mosaic Models for Textures. In A. Rosenfeld, editor, *Image Modeling.* Academic Press Inc., 1981.

[3] M. T. Barnsley. *Fractals Everywhere.* Academic Press., New York, 1988.

[4] C. Bouman and B. Liu. Multiple Resolution Segmentation of Textured Images. *IEEE Trans. Patt. Anal. Machine Intell.*, 13:99–113, February 1991.

[5] A. Bovik, M. Clark, and W. Geisler. Multichannel Texture Analysis Using Localized Spatial Filters. *IEEE Trans. Patt. Anal. Machine Intell.*, 12(1):55–72, January 1990.

[6] P. Brodatz. *A Photographic Album for Artists and Designers.* Dover Publishing Company, 1966.

[7] A. D. Calway. *The Multiresolution Fourier Transform: A general Purpose Tool for Image Analysis.* PhD thesis, Department of Computer Science, The University of Warwick, UK, September 1989.

[8] R. Chellappa and S. Chatterjee. Texture Classification using GMRF Models. *IEEE Trans. Acous. Speech Sig. Proc.*, 33:959–963, August 1985.

[9] R. Chellappa and R. L. Kashyap. Texture Synthesis using 2-D Noncausal Autoregressive Models. *IEEE Trans. Acous. Speech Sig. Proc.*, 33:194–203, August 1985.

[10] G.R. Cross and A.K. Jain. Markov Random Field Texture Models. *IEEE Trans. Patt. Anal. Machine Intell.*, 5:25–39, 1983.

[11] A. R. Davies. *Image Feature Analysis Using the Multiresolution Fourier Transform.* PhD thesis, Department of Computer Science, The University of Warwick, UK, August 1993.

[12] H. Derin and H. Elliott. Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields. *IEEE Trans. Patt. Anal. Machine Intell.*, 9:39–55, 1987.

[13] D. Dunn, W. Higgins, and J. Wakeley. Texture Segmentation Using 2-D Gabor Elementary Functions. *IEEE Trans. Patt. Anal. Machine Intell.*, 16(2):130–149, February 1994.

[14] I. M. Elfadel and R. W. Picard. Gibbs Random Fields, Cooccurrences, and Texture Modeling. *IEEE Trans. Patt. Anal. Machine Intell.*, 16:24–37, 1994.

[15] J. M. Francos, A. Meiri, and B. Porat. A Unified Texture Model Based on a 2-D Wold-Like Decomposition. *IEEE Trans. Signal Processing*, 41:2665–2677, 1993.

[16] A. Gagalowicz. A New Method for Texture Fields Synthesis, Some Applications to the Study of Human Vision. *IEEE Trans. Patt. Anal. Machine Intell.*, 3(5), September 1981.

[17] J. Gårding. *Shape from Surface Markings.* PhD thesis, Inst. of Technology, Stockholm, 1991.

[18] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Patt. Anal. Machine Intell.*, 6(6):721–741, 1984.

[19] R. Gray. Vector Quantization. *IEEE ASSP Magazine*, 1:4–29, 1984.

[20] M. Hassner and J. Sklansky. The Use of Markov Random Fields as Models of Texture. In A. Rosenfeld, editor, *Image Modeling.* Academic Press Inc., 1981.

[21] R. W. Fries J. W. Modestino and A. L. Vickers. Stochastic Image Models Generated by Random Tessellations of the Plane. In A. Rosenfeld, editor, *Image Modeling.* Academic Press Inc., 1981.

[22] A. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Processing*, 1:18–30, 1991.

[23] A. K. Jain. Advances in Mathematical Models for Image Processing. *Proc. IEEE*, 69:502–528, 1981.

[24] A. K. Jain and F. Farrokhnia. Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[25] B. Julesz. Experiments in the Visual Perception of Texture. *Science Amer.*, 232:34–43, 1975.

[26] R.L. Kashyap. Univariate and Multivariate Random Field Models for Images. In A. Rosenfeld, editor, *Image Modeling.* Academic Press Inc., 1981.

[27] J. Keller, S. Chen, and R. Crownover. Texture Description and Segmentation Through Fractal Geometry. *Computer Vision, Graphics and Image Processing*, 45:150–166, 1989.

[28] H.E. Knutsson and G.H. Granlund. Texture analysis using 2-d quadrature filters. In *IEEE CAPADIM Workshop*, Pasadena, 1983.

[29] J. Mao and A. Jain. Texture Classification and Segmentation using Multiresolution Simultaneous Autoregressive Models. *Pattern Recognition*, 25(2):173–188, 1992.

[30] A. Papoulis. *Signal Analysis*. McGraw-Hill, 1984.

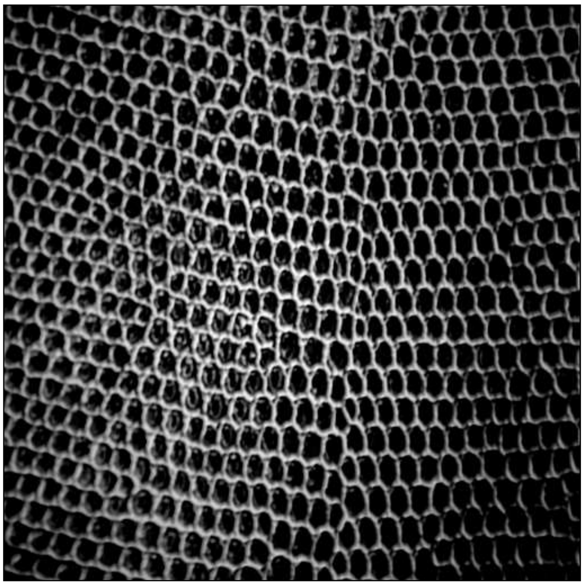
[31] A. P. Pentland. Fractal-Based Description of Natural Scenes. *IEEE Trans. Patt. Anal. Machine Intell.*, 6:661–674, 1984.

[32] M. Porat and Y. Y. Zeevi. The generalized Gabor Scheme of Image Representation in Biological and Machine Vision. *IEEE Trans. Patt. Anal. Machine Intell.*, 10:452–468, July 1988.

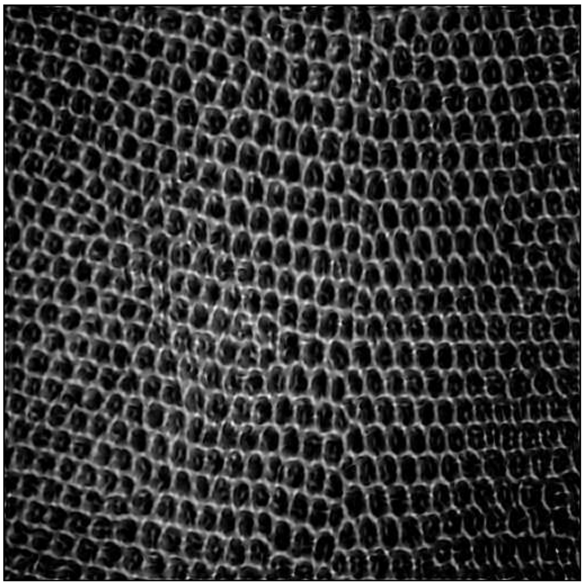
[33] X Serra. A system for sound analysis/synthesis/transformation based on a deterministic plus stochastic decomposition. Research Report STAN-M-58, CCRMA, Department of Music, Stanford University, USA, 1989.

[34] P. Volet and M. Kunt. Synthesis of Natural Structured Textures. In *Proc. of EUSIPCO-86*, pages 913–916, 1986.

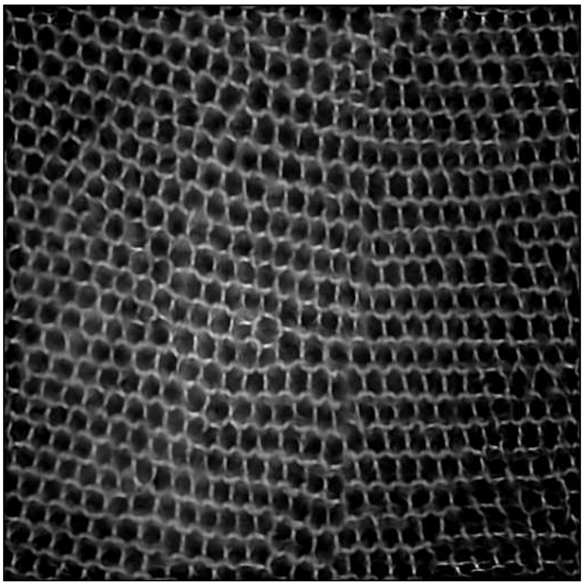
[35] R. G. Wilson and M. Spann. *Image Segmentation and Uncertainty*. Pattern Recognition and Image Processing Series. Research Studies Press Ltd, 1988.



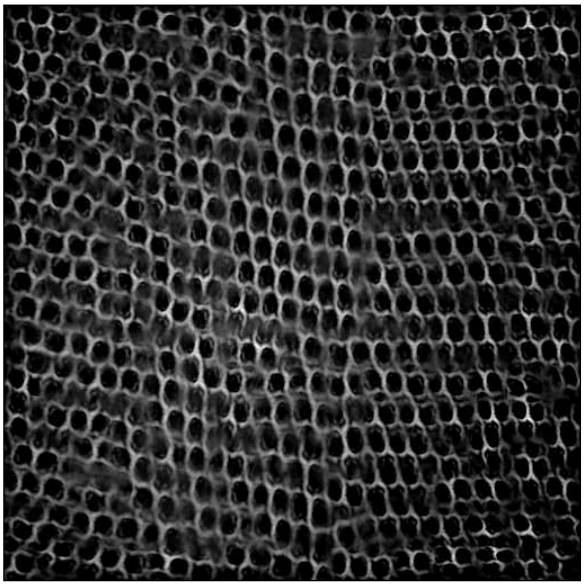
(a) Original Reptile



(b) Synthesis at level 6

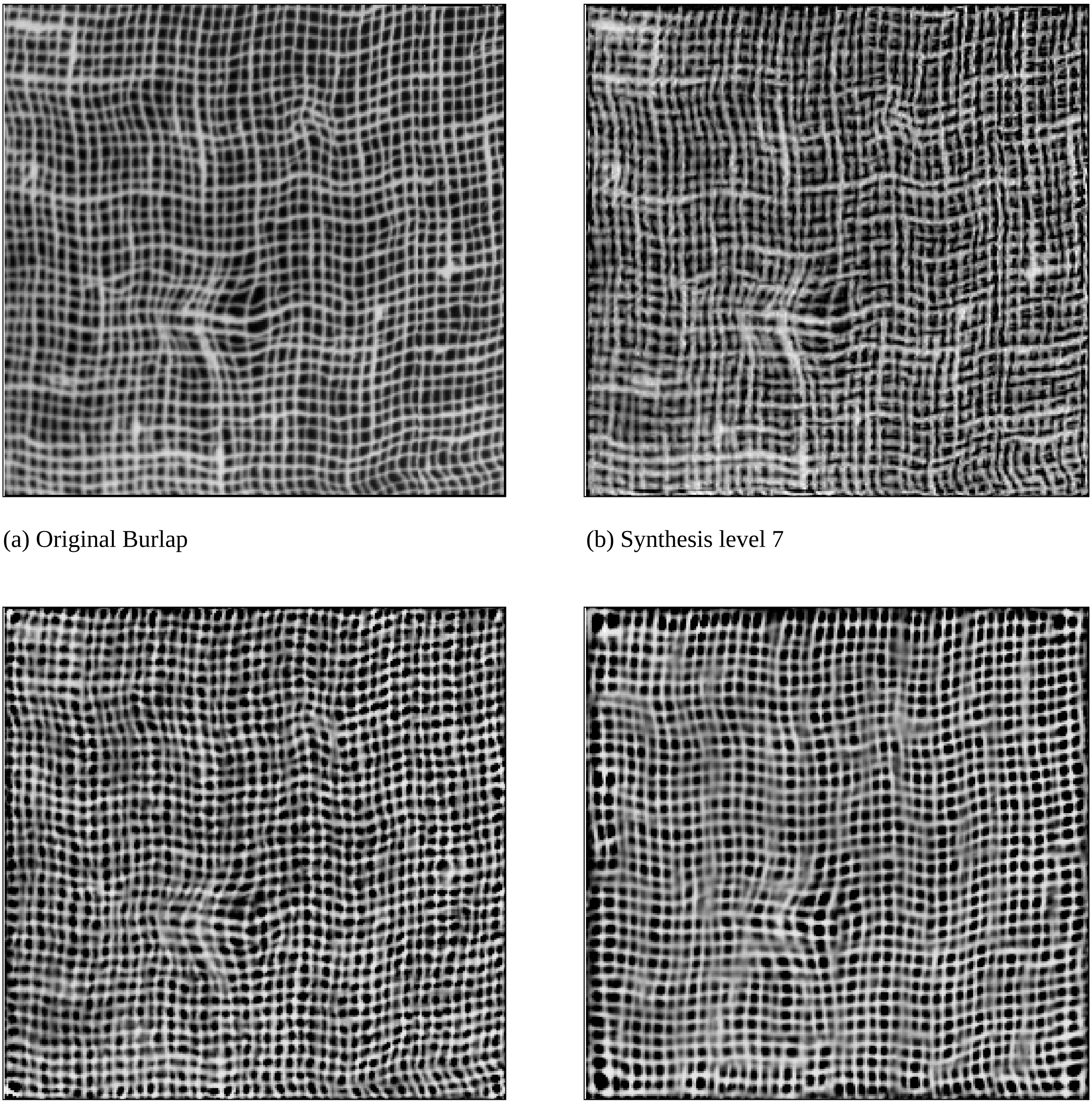


(c) Synthesis at level 5



(d) Synthesis at level 4

Figure 6: Reptile skin image synthesised at different levels.



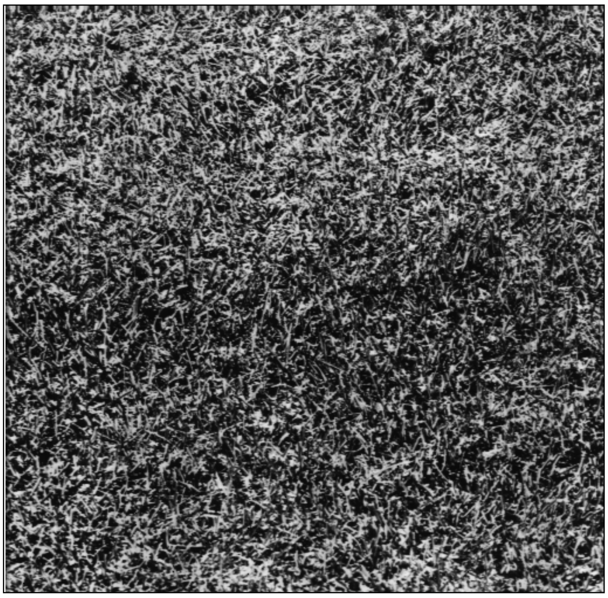
(a) Original Burlap

(b) Synthesis level 7

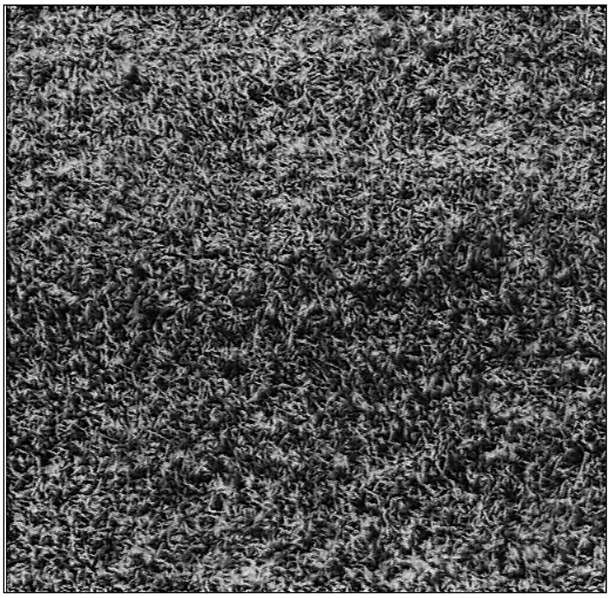
(c) Synthesis level 6

(d) Synthesis level 5

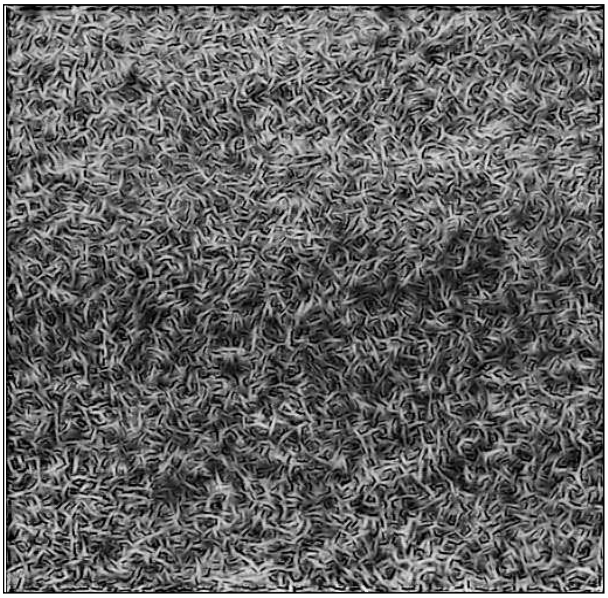
Figure 7: Burlap image synthesised at different levels.



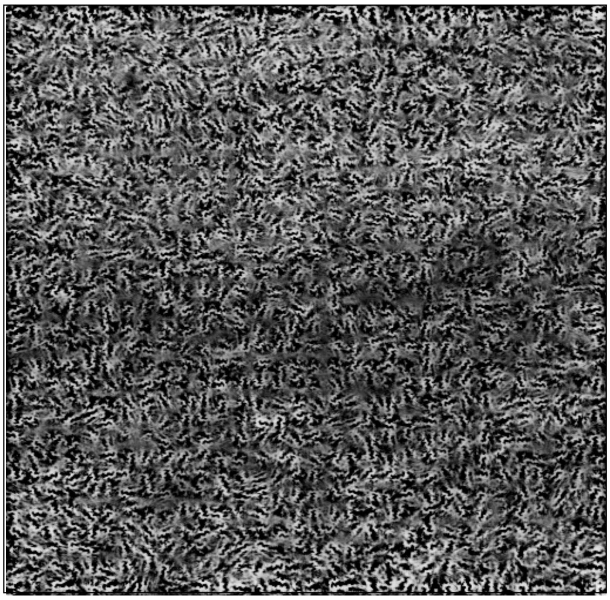
(a) Original Grass



(b) Synthesis level 7



(c) Synthesis level 6



(d) Synthesis level 5

Figure 8: Grass image synthesised at different levels.

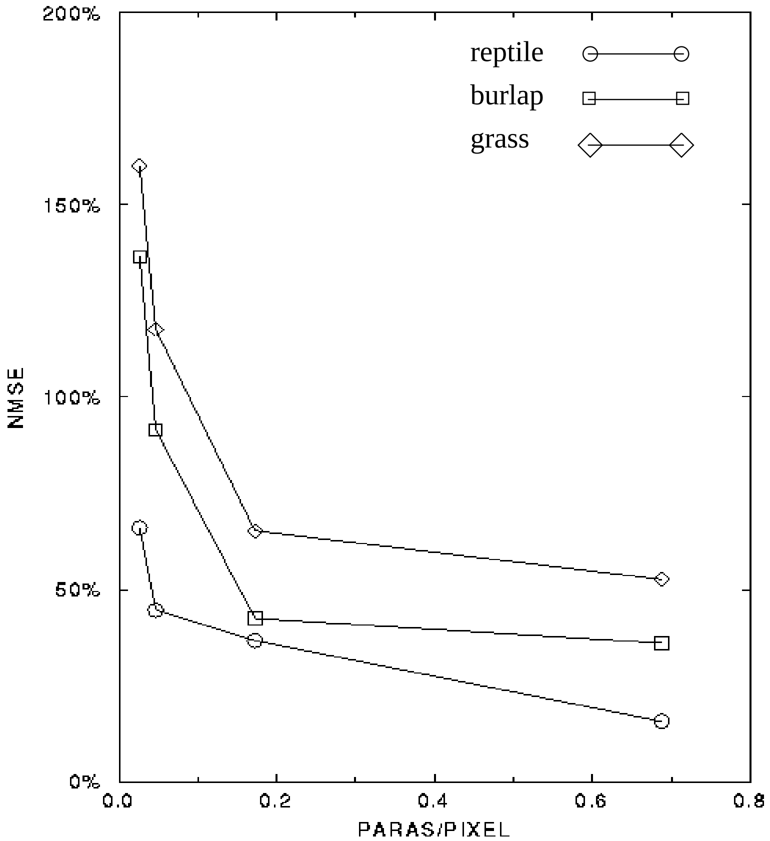
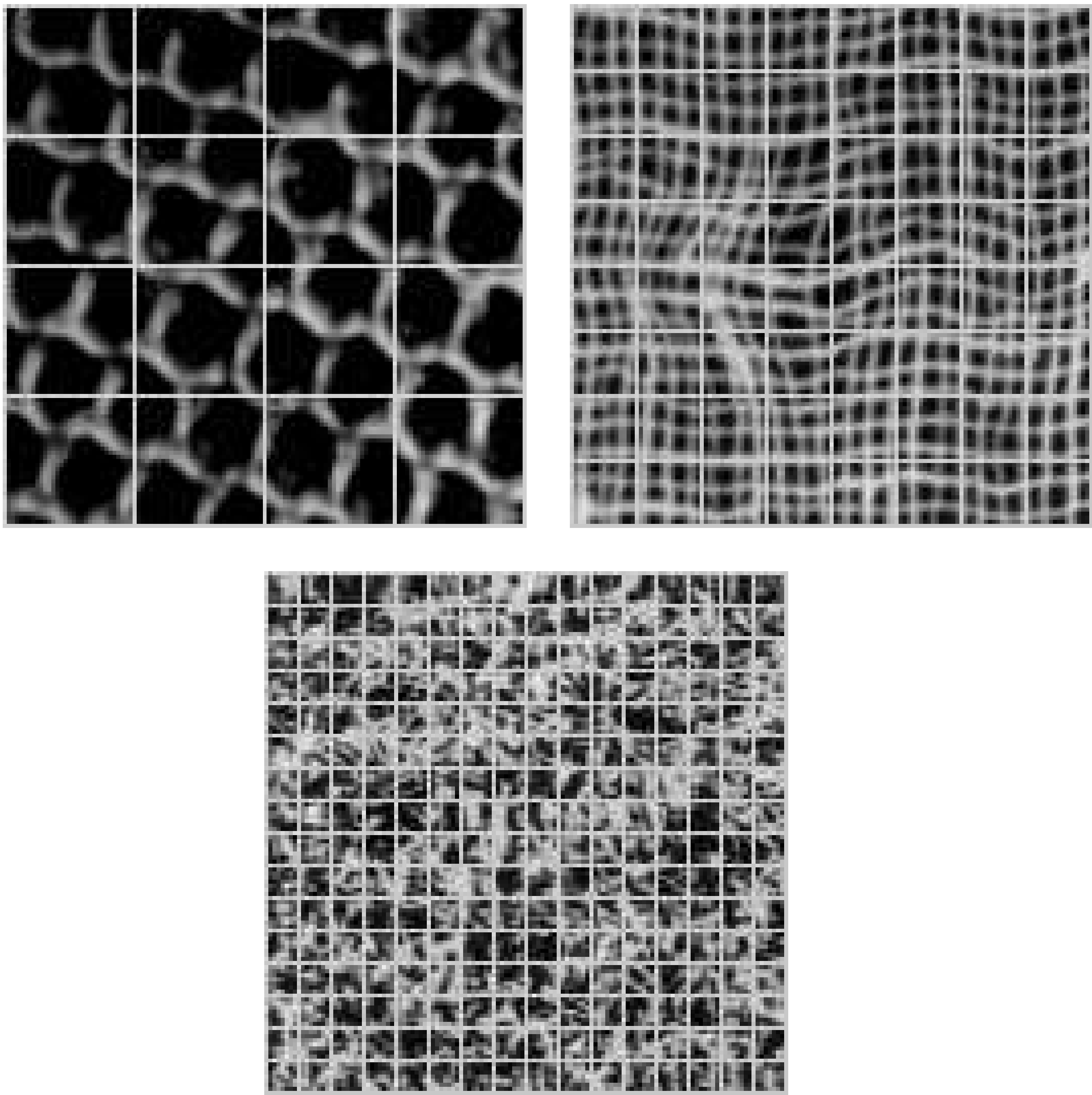


Figure 9: Normalised mean-square error as a function of the number of parameters used for synthesis.

(a) (b)



(c)

Figure 10: The choice of the prototype scale corresponds to different nat ural textural images: (a) reptile skin (b) burlap and (c) grass.