Robust method for texture synthesis-by-analysis based on a multiscale Gabor scheme

Rafael Navarro & Javier Portilla

Instituto de Optica "Daza de Valdés", Consejo Superior de Investigaciones Científicas.
Serrano 121. 28006 Madrid. Spain

ABSTRACT

We propose a new texture *synthesis-by-analysis* method inspired by current models of biological early vision and based on a multiscale Gabor scheme. The analysis stage starts with a *log-polar sampling* of the estimated power spectral density of the texture by a set of 4x4 Gabor filters, plus a low-pass residual (LPR). Then, for each channel, we compute its energy and its two (X,Y) bandwidths. The LPR is coded by five parameters. In addition, the density function of the original texture is also estimated and compressed to sixteen values. Therefore, texture is coded by only 69 parameters. The synthesis method consists of generating a set of 4x4 *synthetic channels* (Gabor filtered noise signals). Their energies and bandwidths are corrected to match the original features. These bandpass filtered noise signals are mixed into a single image. Finally, the histogram and LPR frequencies of the resulting texture are modified to fit the original values. We have obtained very satisfactory results both with highly random textures and with some quasi-periodic textures. Compared to previous methods, ours has other important advantages: high robustness (stable, non iterative and fully automatic), high compactness of the coding, and computational efficiency.

Keywords: texture synthesis-by-analysis, multiscale Gabor filtering, vision models.

1. INTRODUCTION

Texture synthesis is necessary in order to generate realistic synthetic images from a set of compressed data. A common way to imitate real textures is to perform a parametric coding of the original texture, which provides the input to the synthesis stage. Figure 1 shows the *synthesis-by-analysis* (S-A) process.

Figure 1. Texture synthesis-by-analysis block diagram

Traditionally, two main approaches have been taken to study image textures: structural and statistical models[1]. Although real textures are neither completely structured nor purely stochastic, statistical models provide in many cases good visual results, requiring a relatively small set of parameters (specially when modeling natural textures). However, from a general point of view, we could model real textures as a combination of a *stochastical* component with a *structural* component. Here we have adopted a statistical approach, inspired by our current knowledge of the funtioning of the human visual system (HVS).

Many statisticals models have been proposed for modeling non-structured image textures[2]. Fractal theory[3] provides an extremely compact coding and computationally efficient method for modeling the statistical features of many natural images. However, it is clear that the majority of textured images do not even approximately fit the fractal model. Although some extensions of this model have been proposed[4], they have not shown to be general enough to become into a generic tool for texture S-A[2]. Markov models, on the contrary, provide a rich theoretical framework that applies to a very wide set of stochastic textures. However, there are many problems when using them in their general form (not restricted to the linear case), as a base for practical S-A implementations. When using the Gibbs distribution, they require very computationally expensive algorithms and, in addition, there is not a systematic method for choosing the required *energy* function. 2-D adapted Markov chains, on the

 0-8194-2031-X/96/$6.00

other hand, are not suitable for a compact representation of fine quantized gray level textures. Reduced dimensionality variants of this methods yield poor results[2]. However, there are two (overlapping) cases of Markov models that give rise to efficient methods: the autoregressive model (AR) and the multiscale Markov model. These will be discused below.

It has been shown[5] that first and second order statistics are the most important features to visually describe homogeneous purely stochastic textures. A series of methods has been proposed (including the one we introduce here), trying to efficiently model these statistics. Among these, linear models are often used for describing nonstructured textures as 2-D noise signals convolved with linear, spatially invariant (LSI) filters. This modeling problem is basically equivalent to a spectral estimation, where we can either use directly an estimation of the autocorrelation function (or the power spectral density), or calculate the best fitted filter (usually in a least square sense) using autoregressive models (AR), moving average models (MA) or a combination of them (ARMA models). Perhaps, the two most extended methods for stochastic texture modeling are the autocorrelation and histogram (AC-H) models[6,7] and the AR models[8]. AC-H models have derived algorithms to achieve random 2-D signals that simultaneously comply with second and first order statistics specifications. Although interesting from a theoretical point of view and for some statistical simulations, these algorithms are iterative and computationally expensive. Moreover, the convergence of the error to zero is not guaranteed. Visually similar, or even better results can be obtained using other, less rigorous, but much more practical methods (see Section 3). AR models, on the other hand, can in many cases adequately represent the emergent frequencies of the texture, using few parameters, but fail with spectra having zeros near the unity circle in the Z plane. Other drawbacks of AR models are the high sensitivity of the result to the filter size and values, and that, using the causual synthesis method, (i.e., generating a sample from the *previous* ones) one has not full control over the first order statistics. Finally, all the mentioned linear single scale models require the choice of the filter order (or the region of the AC function considered), and, up to now, no robust and general algorithm has been proposed for automatizing this task[9].

Since the last decade, there is an increasing interest in applying multiscale models for image analysis and synthesis, both from a signal processing viewpoint and from a human vision research perspective. Wavelets, multiscale filter banks[10-12], multiscale Markov models[13], etc., are powerful tools for many image processing and vision tasks. Specifically, multiscale schemes are more efficient for coding texture features than single scale models. The high performance of these methods has much to do with the fact that the HVS itself codifies visual information in a multiscale scheme[14]. In a previous work we have proposed a multiscale Gabor scheme, based on the HVS[15], as a common multipurpose first stage for image processing and vision tasks. Here we present a texture S-A method based on this multipurpose scheme. It compares favorably with most extended S-A methods, particularly in terms of robustness and compactness, apart from having a wider range of applications (since it models both random and quasi-periodic textures).

2. GENERAL DESCRIPTION OF THE MODEL

2.1. The multiscale Gabor scheme.

Our multiscale Gabor (MSG) scheme[15] performs a *log-polar sampling* of the image spectrum (here we use the term *sampling* in a wide sense, considering the different spectral regions as *samples* of the whole spectrum), trying to imitate the early vision process[16]. This spectral sampling is done by applying a bank of bandpass Gabor filters with bandwidths proportional to their central frequencies, which are located in a discrete set of radial and angular frequency values. The central frequencies are distributed by octaves, from 1/32 to 1/4 cycles/pixel, while their orientations correspond to multiples of $\pi/4$. Therefore, we have 16 (4x4) filters centered in a log-polar grid (see Fig.2a). As can be seen in Fig.2b, the bandwidths are such that contiguous filters intersect at half-peak height in a radial slice. In order to adequately cover the very low frequencies, an additional low-pass filter is required, corresponding to the *low-pass residual* channel (LPR) that is a Gabor function tuned to the zero frequency.

The actual expression for each (p,q) Gabor filters is:

$$g_{p,q}(x,y) = a_p^2 \exp\left(-\pi a_p^2\left(\left(x\cos\theta_q + y\sin\theta_q\right)^2 + \left(y\cos\theta_q - x\sin\theta_q\right)^2\right)\right) \cdot \exp\left(i2\pi f_p\left(x\cos\theta_q + y\sin\theta_q\right)\right) \quad (1)$$

where

$$f_p = 0.25 \cdot 2^{p-4} \text{ cycles / pixel}, \qquad k = \frac{1}{3}\sqrt{\frac{\pi}{\ln 2}} \quad \text{and} \quad \theta_q = (q-1)\frac{\pi}{4} \text{ radians}. \tag{2}$$
$$a_p = k \cdot f_p$$

The indexes p and q stand for the spectral location (radial and angular, respectively) of the Gabor filter central frequencies. Both indexes range from 1 to 4.

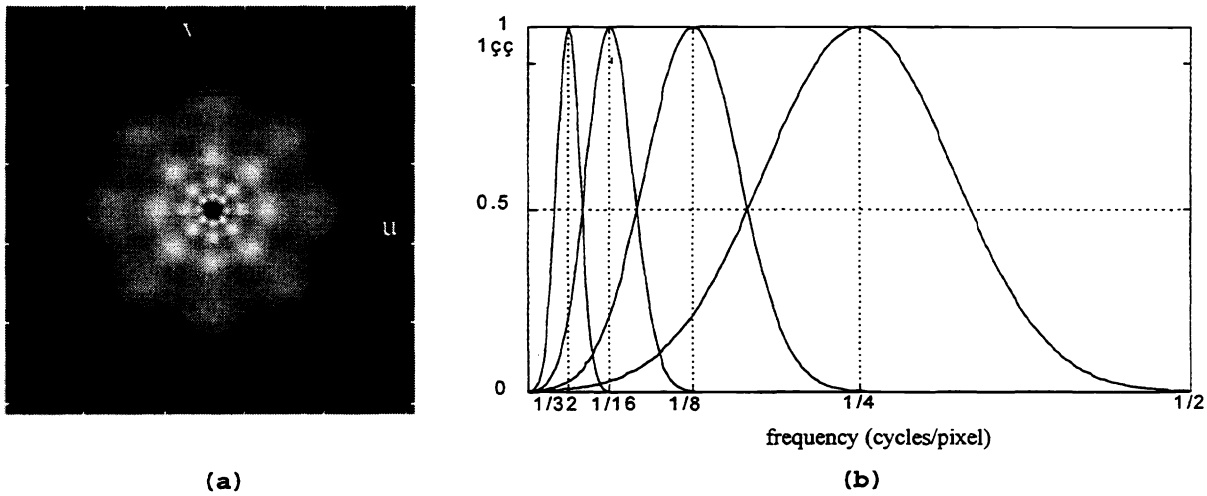

Figure 2. (a) A spectral gray-level representation of the set of Gabor filters. (b) A radial cross-section of the Gabor filters in the Fourier domain.

The main features of this scheme are: It is quasicomplete. Therefore, the original image can not be recovered *exactly* as a linear combination of the 4x4 *Gabor channels* plus LPR. Nevertheless, we can reconstruct an image that is almost visually indistinguishable from the original. It can be obtained by symply adding togheter all the channels. The main source of error is the loss of higher frequencies, which are not considered in our sampling. This produces a low-pass filter effect, that is very small in practice. Finally, the sampling is redundant. As shown in Fig 2, there is considerable overlapping among spectrally adjacent channels. Redundancy slightly increases the complexity of the synthesis proces, but it also increases the robustness of the scheme and is very important for avoiding artifacts.

2.1. Synthesis-by-analysis.

We have applied this scheme to obtain a compact and visually efficient description of the power spectrum of homogeneously textured images. This log-polar sampling is applied to real images, using only even Gabor filters and ignoring the spectrum phase. First order statistics (histogram) are coded independently.

Instead of trying to find a description of the entire power spectrum directly, as do the single scale linear methods, we split the spectrum into 16 Gabor channels, and next, seek for a parametric description of each one of the resulting spectral regions. An important feature of this method is that, as a consequence of the log-polar sampling, the *spectral resolution* of the coded spectrum is approximately constant in a logarithmic scale, decaying quickly (in a linear scale) as the frequency increases. This is an important feature, since the HVS seems to behave in a similar way. Consequently, we can take advantage of this fact to code the spectra of textured images more efficiently in visual terms. Single scale linear models, like those described in the introduction, uniformly *smooth* the power spectrum (or its inverse, in the case of the AR model), and, therefore, they are not optimal from a visual point of view.

This nonuniform sampling of the spectral contents is very useful to code the energy and approximate location of the spectral components of a random texture. However, we have experienced that some additional information about how the energy is spectrally distributed within each Gabor channel is still needed to visually characterize the power spectrum. It is clear that an

image of a sinusoid located at the center of a Gabor channel is visually very different as compared to an image of white noise filtered by the corresponding Gabor filter, although they have the same energy. A similar consideration can be done with other highly concentrated spectra. Thus, to properly code the power spectrum we need a characterization of the visually relevant features of the *spectral shape* of each Gabor channel, in addition to its measured energy. For this purpose we have taken a measure of the degree of spectral concentration of these channels in two directions, as the *equivalent bandwidths* of the Gabor channels. Finally, five parameters are used to code the spectral distribution of the LPR. Then, the 256 gray level histogram is computed and compressed to 16 values. In this way, we obtain a compact encoding with only 48+5+16=69 parameters (see Table 1).

Table 1. Parametric encoding of pure textures

Gabor Channels' Energy	Gabor Channels' Eq. BWs	LPR spectral distribution	Histogram (compressed)
16	32 (2x16)	5	16

In the synthesis process we want to obtain a texture that would yield the same values as the original when submitted to the same feature extraction process. For this purpose we generate a set of independent complex Gaussian distributed white noise 2-D random signals. Each signal is convolved with an elliptical Gabor filter (in practice the image is convolved with a low-pass Gaussian, corresponding to the Gabor filter envelope, and then it is modulated to the central frequency of the Gabor channel). The resulting *synthetic channels* are adequately mixed (see Section 3) to obtain an image. After adjusting the histogram of this image, we obtain the synthetic texture. The following diagram summarizes the entire process.

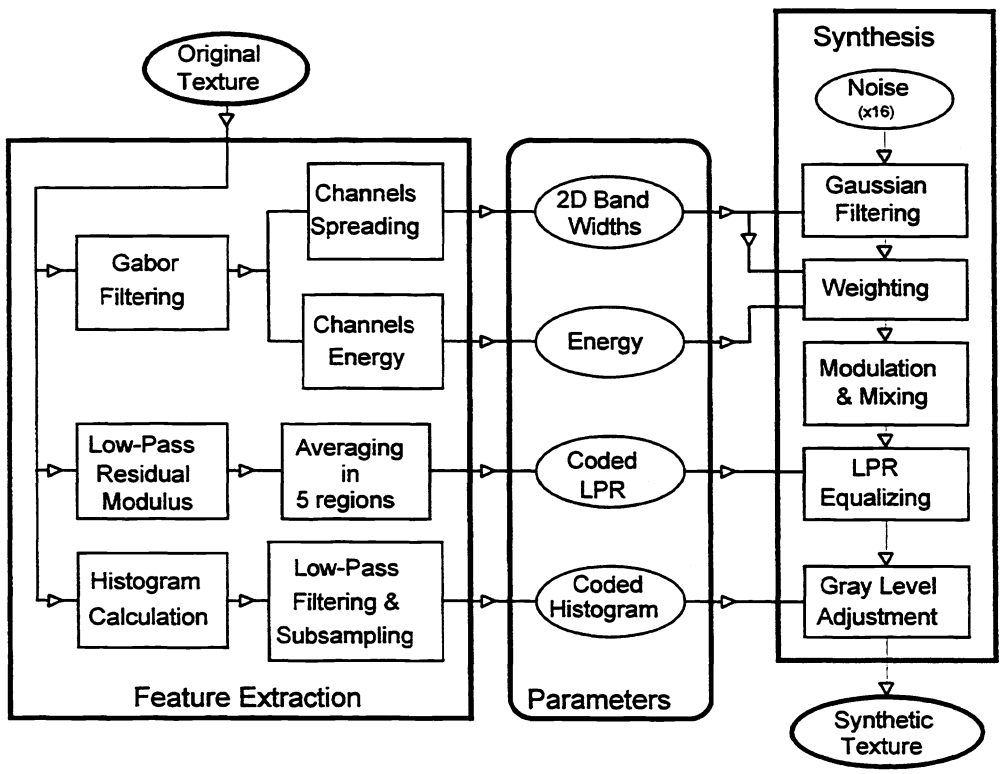

Figure 3. Block diagram of the synthesis-by-analysis method

3. THE S-A METHOD

3.1. Feature extraction.

Due to the global nature of the homogeneous texture analysis, the feature extraction can be done efficiently in the Fourier domain.

3.1.1. Energy of the Gabor channels.

The set of global spectral energies of the 16 Gabor channels provide a first rough sampling of the power spectrum of the textured image. Prevalent orientations and interdistances are reflected in this measure. As an example, Table 2 shows the rms values of the Gabor channels (normalized with respect their mean value) for two Brodatz[17] textures. We can appreciate how this set of parameters reveals very significant information about the texture. A similar, but local, set of parameters has been successfully applied to texture segmentation and classification[12]. Although this parametrization provides very good results, we have experienced that it is far from constituting a visually complete description.

Table 2. Gabor rms values for two Brodatz textures.

	Water				Sand			
	0°	45°	90°	135°	0°	45°	90°	135°
f1	3.5	3.9	2.9	2.2	5.3	6.3	4.6	6.0
f2	5.3	9.9	6.4	5.8	9.3	8.5	9.0	11.3
f3	2.9	6.9	23.3	6.3	12.1	11.1	12.1	14.0
f4	3.3	4.2	15.6	3.8	10.8	9.2	10.6	11.1

3.1.2. Equivalent banwidths of the Gabor channels.

We have applied a method similar to that used in [18] to measure the effective or equivalent bandwitdth of each Gabor channel. In its current implementation, the equivalent bandwidths are calculated over the u and v frequency axes' directions*. To obtain a measurement of the equivalent bandwidth in the horizontal direction, u, we first integrate the energy distribution $P_{p,q}(u,v)$, of the channel p,q:

$$P_{p,q}(u,v)=\left|T(u,v)G_{p,q}(u,v)\right|^2=\left|DFT\left\{(x,y)*g_{p,q}(x,y)\right\}\right|^2 \tag{3}$$

in the vertical direction to obtain $P_{p,q}(u,v)$. $t(x,y)$ is the textured image and $g_{p,q}(x,y)$ is the Gabor filter. Next, we divide the result by its maximum, and finally, the equivalent bandwidth $A_{u_{p,q}}$ is obtained integrating in the horizontal axis (analogously for the v bandwidth):

$$P_{u\,p,q}(u)=\sum_{v}P_{p,q}(u,v),\qquad \overline{P}_{u\,p,q}(u)=\frac{P_{u\,p,q}(u)}{max\left(P_{u\,p,q}(u)\right)},\qquad A_{u\,p,q}=\sum_{u}\overline{P}_{u\,p,q}(u). \tag{4}$$

3.1.3 The LPR channel.

The very low frequencies of a textured image, although not considered commonly as carriers of texture information, are, nevertheless, important and visually significant. In a first approximation we have considered the LPR channel to have significant values within the central spectral region of frequencies u and v, which are smaller than 1/64 cycles/pixel. Fig. 4a shows this region along with the lower frequency Gabor channels (the radius of the circles represents the halfwidth at half height); Fig. 4b shows a detailed diagram of the 7x7 pixels central region for the particular case of a 256x256 pixels texture used in this study.

* This has been done for simplicity. Another possibility is to use radial and angular bandwidths, which would probably be more adequate for most unstructured textures.

The coding consisted of dividing this area into the five represented zones (the DC component, each one of the two axes, and each one of the two square off-axes zones), and then taking the average value within each zone (the spectrum modulus is center-symmetric for real signals, so that we only need to consider one half, plus the center). The location and number of samples of each zone is intended to be roughly adapted to their different visual importance.

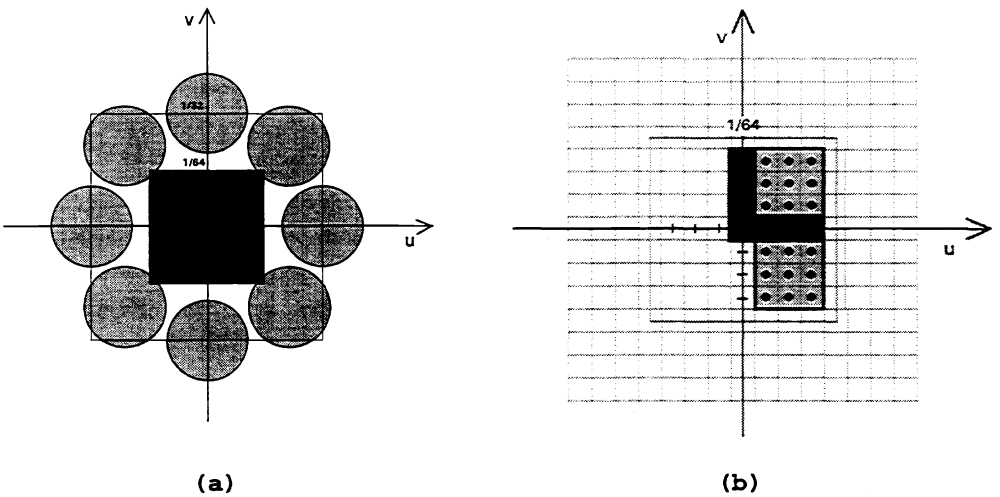

Figure 4. (a) Spectral location of the LPR and the lowest frequency Gabor filters. (b) The five spectral regions of the LPR channel.

3.1.4. Histogram.

First order statistics are visually relevant and they must be considered. In natural textures, the density function tends to be smooth[19]. Therefore, we can get a good approximate of the histogram by using a reduced set of values. Here we have applied a rather crude approximation, consisting of low-pass filtering the 256 values histogram and subsampling it to 16 values. We have applied an odd symmetric lengthening method (see Fig. 4) which, apart from avoiding edge artifacts, has the property of not altering the values of the first and last samples when applying a unity area even symmetric filter. We have verified that no significant differences in the results follow from using the coded histograms instead of the original ones.

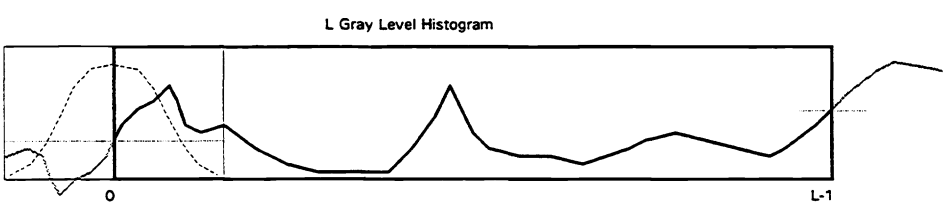

Figure 5. Low pass filtering of the histogram. The dashed line on the left represents a low-pass even filter of unity area. The lighter continuous lines in both sides show the odd symmetric lengthening applied.

3.2. Synthesis.

Sixteen (4x4) independent 2-D signals, consisting of complex white, zero-mean Gaussian noise, are generated, one for each Gabor channel. We force the DFT modulus to be constant at all frequencies, in order to avoid random effects on the spectral overlapping calculations (see next section). We apply a Gaussian low-pass filter, as a convolution kernel in the space domain, to each 2-D noise. Therefore, to transform the Gaussian-filtered signal into a Gabor-filtered one, we modulate it to the central frequency (p,q), and symmetrically replicate its spectrum to produce a real, band-pass signal.

3.2.1. Adjustment of the channels' equivalent bandwidths.

It is straightforward to show that (neglecting small discretization errors and assuming that the spectral values at the highest frequency along both axes are almost zero, i.e. no aliasing) the following 2-D Gaussian signal:

$$g_{eq_{p,q}}(x,y) = K \exp\left(-\pi\left(2A_{u_{p,q}}{}^2 x^2 + 2A_{v_{p,q}}{}^2 y^2\right)\right) \tag{5}$$

where K is an arbitrary constant, would yield the same $A_{u_{p,q}}$ and $A_{v_{p,q}}$ values as the original Gabor channel (p,q). We can apply this expression for designing the synthesis filters from the equivalent bandwidths computed from the original texture. Before doing this, we must keep in mind two phenomena occurring when mixing Gabor filtered noises:

(i) In general, each Gabor channel does overlap with the other synthetic channels of adjacent frequencies. Therefore, the k-th Gabor channel of the synthetic texture, not only contains information from the k-th synthetic channel, but also (to a lower degree), from the spectrally adjacent synthetic channels.

(ii) If we try to apply the multiscale Gabor filtering to the synthetic texture, the effect of each Gabor filter upon the corresponding synthetic channel will be to shorten its bandwidths by a constant factor. This occurs because we are multiplying two Gaussians, and thus, adding their exponents.

The first problem is easy to solve when we want to adjust the energy contents of the Gabor channels (see next Section), but it becomes really complex if we are intended to model the equivalent bandwidths of the resulting Gabor channels as well. Anyway, we have experienced that ignoring the spectral overlapping in the bandwidth adjustment introduces a second order error, which has no relevant visual effects on the synthetic texture. This can be partially explained if we consider that this error would be higher for those channels having a relatively low energy with respect to the adjacent ones, and hence a minor visual significance.

Disregarding the influence of adjacent synthetic channels, the synthesis filters, $g_{s_{p,q}}(x,y$, are obtained from the measured bandwidths by taking into account the effect (ii):

$$g_{s_{p,q}}(x,y) = K_{p,q} \exp\left(-\pi\left(b_{u_{p,q}}{}^2 x^2 + b_{v_{p,q}}{}^2 y^2\right)\right), \tag{6}$$

where $b_{u_{p,q}} = \sqrt{2A_{u_{p,q}}{}^2 - a_p{}^2}$ and $b_{v_{p,q}} = \sqrt{2A_{v_{p,q}}{}^2 - a_p{}^2}$, and $K_{p,q}$ controls the energy of the synthetic channel.

3.2.2. Adjustment of the channels' energy.

Due to the statistical independence of the noise signals used for generating the synthetic channels, the mean square of the result of applying a particular Gabor filter to the synthetic texture can be computed as a weighted sum of the mean squares of the synthetic channels. Therefore, we can express the set of mean square values of the 16 Gabor channels of the synthetic texture as a vector, which is the product of a matrix **R** with the vector of the synthetic channels' mean square values. Thus, we can obtain the later by simply solving the linear system of equations. Using a single index for both the frequency level and the orientation of the channels (e.g.: $n=4p+q$, $m=4i+j$), we can define:

$$e_n^a = \left\langle \left| \tilde{T}_{p(n)q(n)}(u,v) \right|^2 \right\rangle, \qquad e_m^s = \left\langle \left| S_{i(m)j(m)}(u,v) \right|^2 \right\rangle, \tag{7}$$

which are the mean square values resulting from applying the n-th Gabor filter to the synthetic texture and the mean square value of the m-th synthetic channel, respectively. Then we can write:

$$e_n^a = k \sum_{m=1}^{16} \rho_{nm} e_m^s \tag{8}$$

or in matrix form: $\mathbf{e}_s^a = k\mathbf{R}\,\mathbf{e}^s$ (factor k is due to the histogram correction and can safely be ignored). The elements of **R** depend on the degree of overlap between each synthetic channel and each Gabor filter, which, in turn, depends on their bandwidths and relative position in the Fourier plane. It is easy to show that:

$$\rho_{n(p,q),m(i,j)} = \frac{\iint_{u,v} \left|S_{ij}(u,v)\right|^2 \left|G_{pq}(u,v)\right|^2 dudv}{\iint_{u,v} \left|S_{ij}(u,v)\right|^2 dudv}, \tag{9}$$

which, when substituting the expressions for the Gaussian spectra and solving the integrals, yields:

$$\rho_{n(p,q),m(i,j)} = \frac{{a_p}^2}{2A_{u\,p,q}A_{v\,p,q}} \exp\left(-\pi\left(\frac{(u_{a\,p,q}-u_{s_{i,j}})^2}{{A_{u\,p,q}}^2}+\frac{(v_{a\,p,q}-v_{s_{i,j}})^2}{{A_{v\,p,q}}^2}\right)\right) \tag{10}$$

In this expression the u's and v's refer to the central frequencies of the channels and the subindexes a and s indicate Gabor filter and synthetic channel, respectively. Therefore, to obtain the mean square values of the synthetic channels we substitute the vector of measured values on the left side of Eq.7 and solve for $\mathbf{e}^s$. Once we know the mean square value of each synthetic channel, applying $e^s_{n(p,q)} \propto \iint_{u,v} \left|S_{p,q}(u,v)\right|^2 dudv$ and solving the integral, we obtain the weighting factors:

$$K_{p,q} \propto \sqrt{e^s_{n(p,q)} b_{u\,p,q} b_{v\,p,q}}\,. \tag{11}$$

3.2.3. Equalization of the LPR.

The lower frequencies of the resulting signal have not been modeled by the synthetic channels, and thus need to be *equalized*; i.e. changed their spectral magnitude, while keeping their phase. For doing this, we construct a synthetic low-pass residual by replacing all the magnitudes of the LPR frequencies by the measured averages of its corresponding regions in the parameter extraction process. Then we impose a phase continuity by using the Fourier phase resulting from the sum of all the Gabor channels (in practice only the lower frequency channels do contribute), which helps to obtain a realistic aspect in the synthetic texture.

3.2.4. Histogram matching.

Ideally, we would want to simultaneously adjust both first and second order statistics of the synthetic texture. However, there is no easy and exact solution to this. In many cases, the simplest solution consists of adjusting them sequentially. We have adopted this solution, obtaining good results. Using linear methods, the histogram adjustment is necessary to correct the Gaussian density function of the uncorrected synthetic texture. This is the final step of our procedure.

The following equation summarizes the complete synthesis procedure:

$$\tilde{t}(x,y) = \varphi\left(\sum_{p=1}^{4}\sum_{q=1}^{4} K_{p,q}\,\mathrm{Re}\left\{\left[n_{p,q}(x,y) * \exp\left(-\pi\left({b_{u\,p,q}}^2 x^2 + {b_{v\,p,q}}^2 y^2\right)\right)\right]\exp\left(i2\pi f_p\left(\cos\theta_q x + \sin\theta_q y\right)\right)\right\} + r(x,y)\right) \tag{12}$$

where $n_{p,q}(x,y)$ represents the (p,q) noise signal, $r(x,y)$ is the correction term due to the LPR frequencies' equalization and φ is the non-linear function used for adjusting the histogram.

4. RESULTS AND DISCUSSION

Here we present our results comparing them with those obtained with the two of the most used statistical models (the AC-H and the AR). For this purpose we have used Brodatz textures, employing the same numbers of parameters (approximately) to code the texture for the three models. Evaluation and comparison focusing on the visual resemblance of the synthetic texture with the original, but we also evaluate other important features such as robustness and efficiency. Finally, we briefly compare our method with an other multiscale approach reported very recently[11].

The three methods try to model the 1st and 2nd order statistics (estimated by the power spectrum and histogram) of the texture using a random Fourier phase. To establish a close comparison between them, we use the same phase for the three cases, and implement a "generic" S-A scheme, depicted in Fig. 6, which applies for the three models.

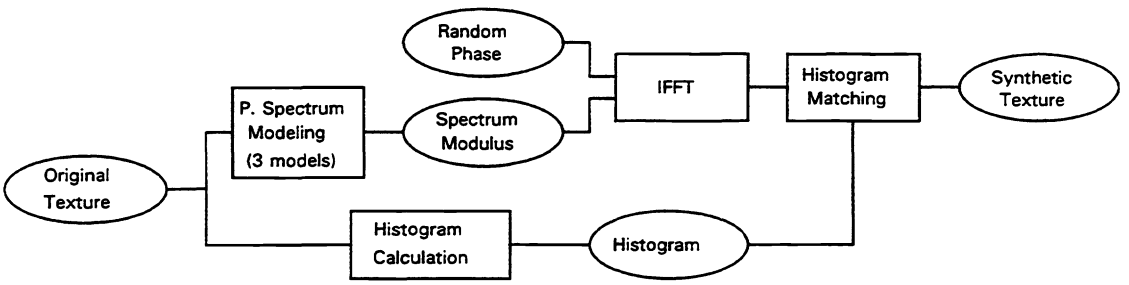

Figure 6. A block diagram of the generic S-A scheme used for the comparative study.

Although the three S-A methods are usually not implemented in this way, it is convenient to use this scheme in order to isolate the differences due to the different coding of the power spectrum from other features of the original S-A methods. In this way, all differences between results are exclusively due to the different coding of the power spectrum. This particular implementation will not affect the visual quality of the results significantly, as compared to more usual implementations. We try to implement a fair comparison, and therefore use the same number of parameters to code the power spectrum for the three methods (actually 53 for our method and a close value, 60, for the other two methods). The AC-H method[6] simply codes the power spectrum by taking a number of samples of the autocorrelation function. We adopt the criterion of implementing the algorithm in its basic form, selecting the central region, as depicted in Fig. 7a. Due to the symmetry of the autocorrelation function, only about half of the samples are needed. The causal autoregresive (CAR), on the other hand, is probably the most widely accepted S-A method. In this case the parameter set is the CAR filter that best models the input texture, in a least squares sense. Fig. 7b shows the set of samples used for the CAR filter.

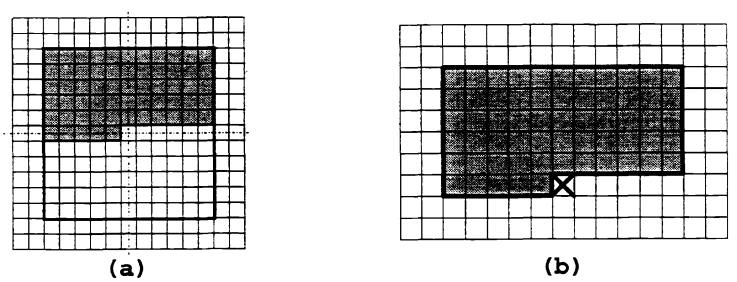

Figure 7. (a) Samples considered to code the autocorrelation in the AC-H method. (b) Same for the CAR filter

Figure 8 shows the results of applying the three S-A methods to five Brodatz textures, plus a text image that we have recorded directly with a scanner. These 6 textures were chosen for having different degrees of structure and regularity, as shown in the first column (left). The second column displays the best-case results that one could expect with this kind of statistical synthetizers. These images were generated using the actual magnitude spectrum of the original texture, and the same random

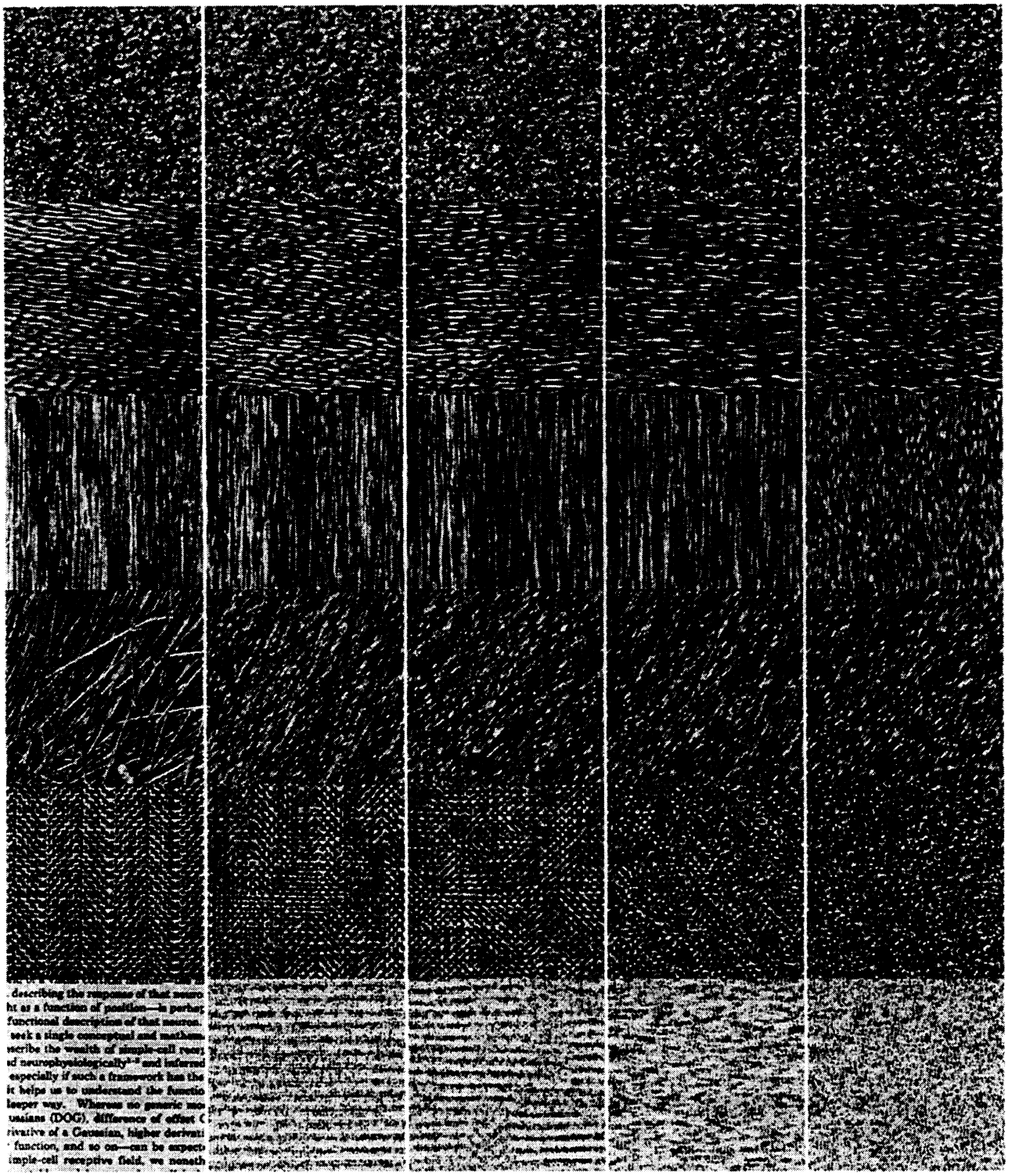

Figure 8. Results of the texture S-A. 1st column shows the originals. 2nd one shows the synthetic textures obtained with the uncoded statistics. 3rd one contains the results of the MSG method (53 parameters). 4th and 5th columns show the results obtained modeling the power spectrum with the CAR and the AC-H method, respectively (both using 60 parameters).

phase than the one used in the rest of cases. The results obtained with our MSG scheme are shown in the third (center) column. The fourth and fifth columns show the results given by the CAR and AC-H models, respectively.

These results indicate that the MSG (center column) provides the best visual results, which for most tested textures are close to the uncoded case (2nd column). The CAR model (4th column) performs similarly in some cases, but it is worse when the texture contains peak frequencies (see cloth texture and text sample). The AC-H model (5th column) yields, by far, the worst results, except for noise-like textures (sand). In general, the three models fail with non regular and highly structured textures, such as straw. Structural elements (straws, sand grains, characters, etc.) are not reproduced, since the original phase information has been ignored. The AC-H model usually does not provide good results using a reduced number of parameters. Of course, we could obtain good results (same as 2nd column) by using the complete autocorrelation function, but then the code would not be compact at all, requiring a number of samples equal to half of the number of pixels in the image. This also applyes to the CAR filter, but the quality will generally be higher for the same number of samples. The MSG model performs the most efficient coding of the second order statistics, producing results visually close to the optimum in all cases (it is remarkable the fact that the MSG compression rate of the power spectrum, for the 256x256 test images, is greater than 600).

The differences between the optimum and the MSG results are mainly due to the spectral shift of the textures' emergent frequencies toward the nearest central frequencies of the MSG scheme. This is more apparent for the water texture, where the waves, slightly tilted in the original, become horizontal in the synthetic image. In the text sample, the frequency shift makes that the 14 lines in the original become 16 in the synthetic image (it is interesting that the eye does not notice this change, unless one proceeds to the task of counting lines). In addition, the resulting text lines are not perfectly clean and straight, mainly because a spectral peak broadening effect due to the particular implementation of our coding process (this problem may be highly reduced by a more accurate coding of bandwidths).

In addition to providing high quality results using a small set of parameters, the main feature of the MSG method is robustness, especially when compared with other methods. First, it uses a fixed number of parameters, no matter the size or features of the input image. Other linear single-scale models (AR, MA, AC-H, etc.) require adapting the number of parameters to the characteristics of each texture for an efficient coding. In addition, our MSG method does not require making any decision, such as choosing the shape of the CAR filter, or about the significant part of the autocorrelation function. Therefore, the MSG method is fully automatic. Contrary to other methods, MSG is inherently stable. The parameters are robust to noise, and can be coarsely quantized. For example, the results shown in Fig. 8 were obtained using only five quantization levels for the bandwidths. The MSG method is not iterative and the amount of computation is invariant, thus permitting an exact time scheduling. Finally, all the involved operations (except the histogram matching) are linear. The analysis can be efficiently implemented in the Fourier domain, and the synthesis in either space or frequency domains. Most of the computation comes from applying the bank of analysis and synthesis filters (using an efficient pyramid implementation) and from the histogram matching.

Finally, another visually inspired, multiscale texture S-A method has recently been reported by another group[11], which shares several basic features with our MSG model. One of the main differences between both models is that they apply an iterative histogram matching method, which adjusts the histograms of each channel and of the whole image (this results in less robustness and more computation than our method). In addition, it does not propose any compact parametric coding of the texture. This method produces very good results with some locally inhomogeneous textures (due to the individual adjustment of the histograms' channels), but fails with quasi-periodic textures (due to the lack of control of the channels' bandwidths).

5. ACKNOWLEDGEMENTS

We are thankful to M. Rynders for style suggestions on the final wording of this work. This work has been supported by the CICYT (Spain) under grant TIC/94-0849.

6. REFERENCES

1. R.M. Haralick, "Statistical and structural approaches to texture", *Proc. IEEE*, vol. 67(5), pp.786 (1979).
2. H. Ivernen, T. Lonnestad, "An evaluation of stochastic models for analysis and synthesis of gray-scale texture", *Pattern Recognition Letters*, vol.15, pp.575 (1994).
3. B.B. Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman and Co., San Francisco, CA (1982).
4. Lewis, "Generalized Stochastic Subdivision ", ACM Trans. Graphics, vol. 6, pp.167 (1987).
5. B. Julesz, "Visual pattern discrimination", *IRE Transactions on Information Theory*, vol. 8(1), pp.84 (1962).
6. A. Gagalowicz and S.D. Ma, "Sequential synthesis of natural textures", *Computer Graphics and Image Processing*, vol. 30, pp.289 (1985).
7. S.D. Ma, A. Gagalowicz, "A parallel method for natural texture synthesis", *Proc. on 7th Int. Conf. on Pattern Recognition*, vol. 1, pp.90 (1984).
8. J.W. Woods, "Two-Dimensional Discrete Markovian Fields", IEEE Transactions on Information Theory, vol.18, pp.232 (1972)
9. C.W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, pp. 548-549, Prentice Hall, NJ 07632 (1992).
10. D. Cano & T. Ha. Minh, "Texture Synthesis Using Hierachical Linear Transforms", Signal Processing, vol.15, pp.131 (1988).
11. D.J. Heeger, J.R. Bergen, "Pyramid-Based Texture Analysis/Synthesis". Computer Graphics. Proceedings of ACM. SIGGRAPH '95 (1995).
12. R. Navarro, O. Nestares, J. Portilla, A. Tabernero, "Several Experiments on Texture Analysis, Coding and Synthesis by Gabor Wavelets", Publicaciones del Instituto de Óptica "Daza de Valdés", vol.52 , Madrid (1995).
13. M.R. Luettgen, W.C. Karl, A.S. Willsky and R:R: Tenney, "Multiscale Representations of Markov Random Fields", IEEE Transactions on Signal Processing, vol.41(12), pp.3377 (1993).
14. A. Rosenfeld, *Multiresolution Image Processing and Analysis*, Springer-Verlag, New York/Berlin (1984).
15. R. Navarro, A. Tabernero, "Gaussian Wavelet Transform: Two alternative fast implementations for images", *Multidimensional System and Signal Processing*, vol.2, pp.421 (1991).
16. R. Navarro, A. Tabernero, and G. Cristobal, "Image representation with Gabor wavelets and its applications", to be published in *Advances in Imagin and Electron Physics*, P.Hawkes, de. Academic Press, San Diego, CA (1996).
17. P. Brodatz, *Textures: A photographic album for artists and designers*, Dover, NY (1966).
18. B. MacLennan, "Gabor representation of spatiotemporal visual images", *Technical Report CS-91-144*, Computer Science Department, University of Tennessee (1991).
19. D.L. Ruderman, W. Bialek, "Statistics of natural images: Scaling in the woods", *Physical Review Letters*, vol.73, pp. 814 (1994).