Volume xxx, (1999) number yyy pp. 000–000

Anisotropic Solid Texture Synthesis Using Orthogonal 2D Views

J.M. DISCHLER, D. GHAZANFARPOUR and R. FREYDIER

Laboratoire MSI
E.N.S.I.L. – Université de Limoges
TECHNOPOLE
87068 Limoges Cedex – France

Abstract

Analytical approaches, based on digitised 2D texture models, for an automatic solid (3D) texture synthesis have been recently introduced to Computer Graphics. However, these approaches cannot provide satisfactory solutions in the usual case of some natural anisotropic textures (wood grain for example). Indeed, solid texture synthesis requires particular care and sometimes external knowledge to "guess" the inside structure of solid texture because only 2D texture models are used for analysis. By making some basic assumptions about the internal structure of solid textures, we propose a very efficient method based on a hybrid analysis (spectral and histogram) for an automatic synthesis of solid textures. This new method allows us to obtain high precision solid textures (closely resembling initial models) in a large number of cases, including the difficult case of anisotropic textures.

Keywords: realistic texturing, anisotropic solid texture synthesis, spatial and spectral analysis, filtering, histogram.

1. Introduction

Textures are omnipresent in the real world and their perception represents an important characteristic of the human visual system. Therefore, texturing has a great importance for rendering realistic images and it has been extensively studied in Computer Graphics. Solid (3D) texturing was mainly introduced by [1] and [2]. There are multiple advantages for such an approach compared to the conventional 2D texture mapping: the application of solid textures to any kind of 3D objects, even in the case of complex ones (fractal objects for example), is straightforward without any texture discontinuities and deformations on objects' surfaces. Therefore, solid texturing represents a widely used and important concept for realistic image rendering.

Solid texturing remained for a long time an experimental domain in which textures were defined as direct procedures throughout the 3D space [3]. The major problem with these approaches is the determination of "appropriate" procedures parameters in order to achieve desired types of textures. We have introduced in Computer Graphics the first method for an automatic generation of solid textures [4] by using the Fourier transformation of a digitised 2D texture model. Our spectral analytical approaches [4, 5] as well as the pyramidal (wavelet) analytical approach of [6] tend to resolve the problem of solid texture parameters definition of classical methods. In spite of an automatic solid texture generation, these approaches are more or less limited. [6] cannot be used in the case of anisotropic textures (see section 2), although this case cannot be neglected because it concerns a great number of usual natural textures (see figures 1, 5 and color plate 1 for some examples). Though our previous approaches can deal with anisotropy, there may be problems of texture variations in [4] or method convergence with [5] in some cases (see section 2).

The main goal of this paper can be stated as fol-

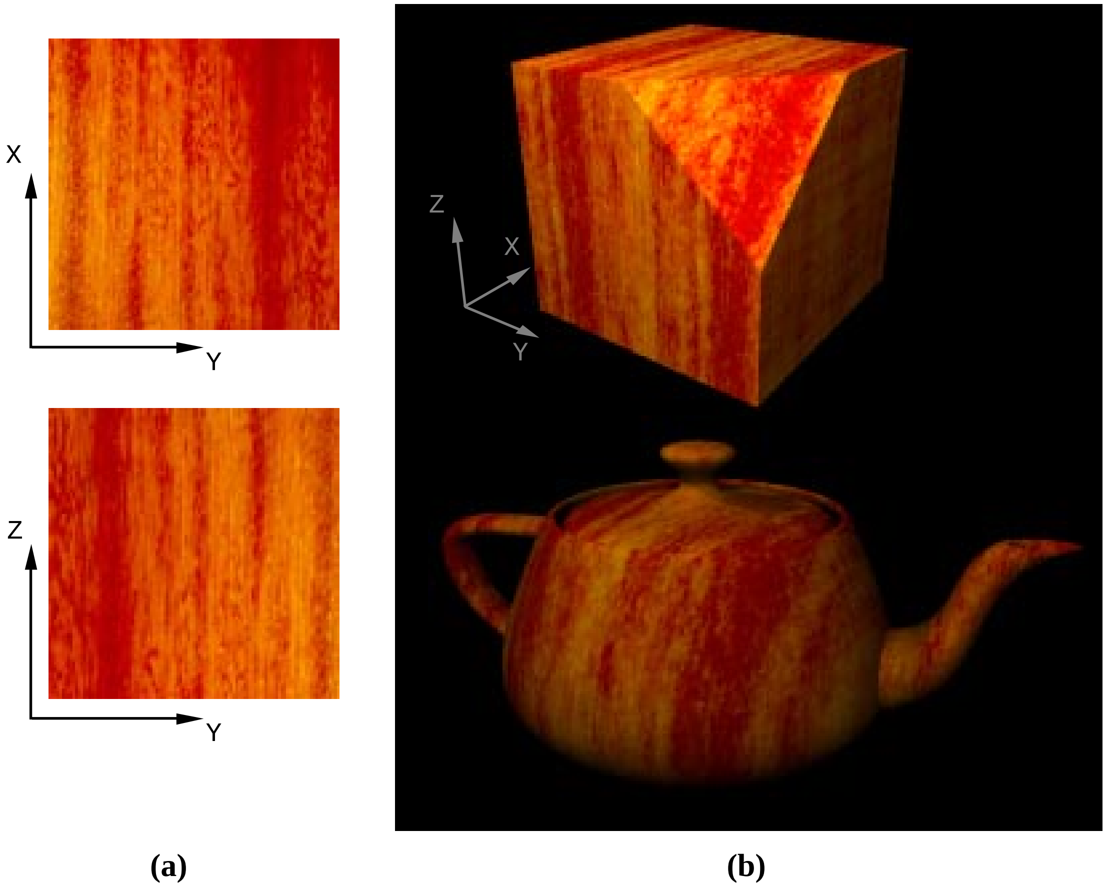

Figure 1: *An anisotropic natural wood grain texture. (a) digitised 2D texture models representing different orthogonal views of the natural texture; (b) the synthetic solid texture obtained by using the method developed in this paper.*

lows: generating a solid texture bloc, defined as a N^3 array of voxels, by using only some digitised 2D models, representing different orthogonal views of a natural solid texture. This means developing a high quality "2D texture analysis for 3D synthesis" method suitable for a great number of different textures including anisotropic ones. This goal is beyond the possibilities of the above mentioned methods [4, 6, 5]. For example, figure 1 illustrates a solid texture synthesis using the method that will be introduced in this paper. On the left, we have supplied two digitised texture models representing two orthogonal views of a natural wood grain texture. On the right, a solid texture, defined as a set of voxels (top image), was constructed and rendered on the surface of a teapot (bottom image). This new method allows a user to supply several views of a texture for generating a solid texture, in a reliable and intuitive way, that closely matches the digitised models.

In the following pqrts of this paper, we first discuss some related works and present our motivation. Then we state the problem mathematically, by introducing the equation that we need to solve for generating the solid textures. We also present the framework of our method, and mainly that of stationary, non structured, texture fields. In section 4, we introduce the slice process, which is based on a spectrum and histogram matching procedure. In section 5, we describe the whole method and present some results. In fact, this method significantly extends and improves our previous one [5]. In this paper, we more accurately address a wider range of textures at considerably lower computational requirements. This new method represents a first step in this domain. Before concluding, we discuss some inevitable limitations, due to the extreme difficulty of using 2D analysis for 3D synthesis.

2. Related works

In the framework of analytical methods for solid texture synthesis, we first need to study the human texture perception mechanisms in order to get a kind of "quality" measurement (how good is the visual resemblance between the synthesised texture and the digitised model). Some human "early vision and texture perception" experiments [7] have shown that the most important characteristics of human texture vision are predominantly local low order statistical properties, such as positions, shapes and brightness of the textural elements called textons [8]. Low order statistics of the positions and the shapes are related to the frequency domain, and brightness is often related to the histogram. Indeed, both considerations (frequency domain and histogram) yielded some texture feature extraction and classification methods such as [9, 10] for frequency domain and [11] for histogram. Researchers have converged to the theory that the human visual system performs an analysis that is based on a bank of space-localised band-pass filters, which allows a very robust human preattentive texture segmentation ability [12, 13]. Gabor elementary functions (or wavelets) allow a joint space-frequency representation, and match the human visual perception theory. For this reason, Gabor wavelets were extensively used in the field of texture segmentation [14, 12, 13].

Some previous 2D texture analysis and synthesis approaches used a unified structural (for example spectral based) and statistical (for example autoregression based) model [15]. At that time, solid textures generation was not addressed in analytical methods. Indeed, it is very difficult to digitise directly a 3D sample of a natural solid texture. Consequently, the use of 2D texture models for analysis seems to be inevitable even in the case of solid textures. The problem of automatic solid texture synthesis using 2D texture analysis was addressed for the first time in [4]. Unfortunately, this seminal method has a serious restriction: only 2D variations are controlled properly because only one 2D texture model is analysed. In fact, generated solid textures represent "perturbed" 3D extensions of 2D textures.

The pyramidal texture analysis-synthesis method of [6] uses the mentioned visual perception theory for texture synthesis. Though more particularly focusing on the synthesis of 2D textures, this method was also used to synthesise solid textures by extending isotropically the pyramids elements to a higher dimension. Unfortunately, the method falls short if more general 3D cases are considered. Indeed, only rather noisy and highly isotropic textures can be processed. The method fails for example in the case of anisotropy as for wood grain. In addition, supplying only one 2D model for analysis is obviously insufficient to catch all aspects of the solid texture.

Because of restrictions due to the use of only one 2D texture model, we have introduced in [5] a method that allows one to supply multiple 2D models for synthesis. The models represent different orthogonal views of a natural solid texture. This method, based on 2D spectral analysis, supposes that few models are representative enough to characterise the "inside structure" of the solid textures. In fact, it supposes that the 2D visual aspects of parallel slices taken from the 3D texture are invariant by displacement along all axes. This is not a supposition of isotropy, since isotropy means that statistical properties are invariant by rotation. Using this "visual invariance" assumption along some axes, a relaxation process is applied for progressively building a synthetic solid texture that matches more and more closely the visual resemblance of the 2D models. The principle consists of "slightly" modifying all slices in a "slice process" to make them correspond more and more to the visual characteristics of the respective models. This method works well in some cases, but as we have mentioned in [5], it can only be applied to a limited number of texture types because of the problem of phases in the slice process (accomplished in the frequency domain). In fact, the phases process supposes that there exists a simple correlation among the phases of the respective models [5]. This correlation is defined by a displacement, that consequently makes the slices very close to the models, if the phases are made very close. Therefore, the relaxation process may not converge to a stable solution or produces unsatisfactory visual results in many cases. On the contrary, if the phases are made very far off from the model phases, the visual resemblance will be deteriorated, except in the case of textures that can be entirely characterised by the energy distribution in the frequency domain such as ocean waves for example.

Very recently a new technique based on a multiresolution (Laplacian and Gaussian pyramidal) analysis was proposed in [16]. This method consists of "randomly exchanging" at each level of the pyramid, elements that can indeed be exchanged without altering the visual result. The choice of elements exchanging or not is based on an error measurement (square difference) of local filters outputs (mainly edge detecting filters) applied to parent elements (at higher levels in the pyramid)[16]. This method captures textural information very well, but it cannot be used for our context of solid textures, since basically the synthesis remains entirely 2D.

To conclude this discussion, we note that there is in fact no efficient existing method that allows users to build "general" (including anisotropic) solid textures from digitised views. The purpose of this paper is to extend and highly improve our previous method [5] in order to process stationary texture fields efficiently. Our new approach is free of the above mentioned restriction, since phases are not considered (see section 4). It yields final visual results of much better quality at considerably lower computational requirements. The convergence is reliable and in addition, there is an obvious resemblance among the digitised models and the synthesised solid textures. This new method can be considered as a first efficient approach that allows us to build "intuitively" and completely automatically a wide range of different solid textures, including anisotropic ones.

3. Mathematical formulation of the problem

3.1. Basic statements

We suppose that the analysed texture fields correspond more or less to stationary random fields with no particular macro-structures inside (precise geometric shapes). This condition is met for many natural textures. The more difficult problem of macro-structures will be discussed in section 6. Due to stationarity, global statistical properties are invariant by displacement. For this kind of textures, a spectral analysis approaches is well adapted. Thus, we believe that a usual spectral analysis is an appropriate tool for extracting many features such as regularity, noise or strong directionality (the latter can only hardly be recovered using pyramids, even if steerable, as the case of wood grain illustrates this). The gaussian spatial windowing of Gabor wavelets, that are successfully applied in the field of texture discrimination and segmentation becomes un-necessary in the case of stationarity, since the global energy distribution in the frequency domain does not change much. In the frequency domain, we use additional statistics to measure the typical amplitude variance according to frequencies. This allows a decomposition of the domain into regions corresponding to the different mentioned features. The statistics are obtained by selecting different sub-parts from the models.

However, the frequency domain alone is not sufficient to capture a sufficient amount of texture properties, even if additional statistics are used and even if the texture field is stationary, since this is only equivalent to a global autocorrelation in the spatial domain. The mentioned vision experiments [7] indicate that brightness (or in a multi-spectral case colour) is also important. The brightness is related to the histogram. Thus, we shall suppose that many stationary texture fields (still containing no random shape-structures or sparse precise visual information) having both same spectral energy distribution and colour histogram are visually "similar".

3.2. The equation

In order to express mathematically the equation that corresponds to the solid texture block that we finally want to synthesise, we introduce some basic operators. We first call V a visual resemblance operator. Two texture pictures I_1 and I_2 represent the "same" texture, if $V(I_1) = V(I_2)$. Let be FDE and H respectively a frequency domain energy and an histogram resemblance operator. Because of the stationary nature of the texture fields, and according to the texture perception experiments made in [7], we shall suppose that:

$$V(I_1) = V(I_2) \equiv \left\{ \begin{array}{c} FDE(I_1) \simeq FDE(I_2) \\ and \\ H(I_1) \simeq H(I_2) \end{array} \right.$$

where $\equiv$ means equivalent to. We call M_{xy}, M_{xz} and M_{yz} the orthogonal models and B_{xyz} the solid texture block, composed of an array of N^3 voxels $v(i,j,k)$. On each voxel v of the 3D block, we can define three orthogonal slices containing v. We call them $s_{xy}(v)$, $s_{xz}(v)$ and $s_{yz}(v)$. A 3D texture block matches the assumption of visual invariance along the three axes if :

$$\forall v \in B_{xyz} \left\{ \begin{array}{c} FDE(s_{xy}(v)) \simeq FDE(M_{xy}) \\ H(s_{xy}(v)) \simeq H(M_{xy}) \\ FDE(s_{xz}(v)) \simeq FDE(M_{xz}) \\ H(s_{xz}(v)) \simeq H(M_{xz}) \\ FDE(s_{yz}(v)) \simeq FDE(M_{yz}) \\ H(s_{yz}(v)) \simeq H(M_{yz}) \end{array} \right.$$

This is basically the formal expression of the equation that we must solve. It is not a linear system, since it uses a convolution operation (because of the frequency domain). Additionally, the equations are expressed on intervals according to the resemblance measurements (it depends on the authorised range) of the operators FDE and H. There exists an infinite number of solutions and, obviously, we cannot resolve this system directly.

The solution we propose consists, as for [5], of using a relaxation process, i.e. a progressive refinement resolution approach. The principle consists of starting with a white noise block (making sure that all frequencies are represented), and then processing all slices to coerce their visual resemblance to match the respective models ones. Unlike the method of [5], we use a different sequence, which makes the system converge faster (see section 5).

4. Coercing 2D slices to match the visual aspects of a model

In this section, we discuss the slice process. Each slice of the 3D block corresponds to one 2D picture of size N^2. The slice process is therefore a 2D process. It is composed of two steps: a spectral step, consisting of coercing the amplitude values of the slice in order to match the models whole energy distribution and a spatial step consisting of making the spatial histograms of the slice match that of the model. We first expose the spectral step then the spatial one.

4.1. Spectral matching

In practice, we dispose of coloured texture images (the models) from which we extract q sub-images (sub-models) of sizes equal to 128*128, 256*256 or 512*512 according to the wanted final texture block size N. This allows us to deal only with statistics in the frequency domain. Generally, we select the sub-models randomly out of one model, as depicted in figure 2, but still in order to cover at least all parts of it. We assume that the sub-models are large enough to satisfy the stationarity criterion and maintain enough textural information. For example a sub-model taken from a wood grain texture model should cover at least a few periods of veins in order to be able to accurately characterise this wood. Colour is either processed by colour quantification, but this works only well in the case of textures characterised by one unique predominant colour (such as the typical brown colour in the case of wood) or can be processed by recomputing the "rgb" components into a decorrelated basis as suggested in [6]. In the later case, each component can be processed independently.

With the help of the sub-models, we compute a certain number of statistical values that allow us to characterise the texture. Now, intuitively the objective of this technique is to modify the amplitudes of a slice in order to achive the same global statistics (mainly the same global energy distribution) as the model in the frequency domain. This corresponds to the texture vision theory mentioned in section 2. Therefore, for q crops, calling $F(f,g)$ the fast Fourier transform (FFT) of one sub-model, we compute:

- the average amplitudes:

$$F_{avg}(f,g) = \frac{1}{q}\sum_{k=1}^{q} |F_k(f,g)|,$$

  where $||$ represents the amplitude of a complex number;
- the maximum and minimum amplitudes :

  $F^+(f,g) = \max_k |F_k(f,g)|, and\ F^-(f,g) = \min_k |F_k(f,g)|$

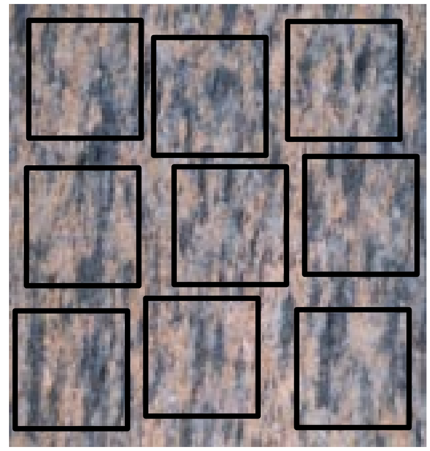

Figure 2: *Different sub-models extracted from a digitised image (model) of a marble-like texture field. The sub-models allow us to compute statistics in the frequency domain.*

The values of F^+ and F^- permit us to characterise the typical range of a frequency component (f,g) of the FFT. The average F_{avg} is used to coerce the corresponding components of the slice to match the determined typical range (see below).

Individual statistical values of the frequency components cannot account for the entire energy distribution in the frequency domain. Therefore, a partitioning of the domain into regions was implemented. The principle consists of grouping frequencies with similar amplitudes. Equalising the energy of each region between the model and the slice, then ensures a similar global energy distribution.

In a first step, it is necessary to classify the energies according to regions. Therefore, we use all $F_{avg}(f,g)$, whose values determine an interval $I = [A,B]$, where $A = \min(F_{avg})$ and $B = \max(F_{avg})$. I is subdivided into z sub-intervals $I_1 = [A_0, A_1[$, ..., $I_k = [A_{k-1}, A_k[$, ..., $I_z = [A_{z-1}, A_z]$, where $A_0 = A$ and $A_z = B$. Generally, it is not convenient to take a constant length for the intervals I_k, as in practice, the picks of FFTs are usually high if compared to the rest of energy. This results in high energy concentrations in only a low number of sub-intervals, whereas the majority remains "empty". For this reason, logarithmic scales are frequently used to represent FFTs rather than linear scales. Hence, it is convenient to use also constant interval lengths, but in a logarithmic scale, i.e. $\log(A_k) - \log(A_{k-1}) = (\log(A) - \log(B))/z$. The subdivision yields a partition of I.

In the second step, we now refer each frequency (f,g) by a number $z(f,g) \in [0, z-1]$, such that

$F_{avg}(f,g) \in I_z(f,g)$. Thus, we also obtain a partition of the frequency domain into regions J_z, where J_z is given by:

$$J_z = \{(f,g) \in [1,N]^2 \;/\; z(f,g) = z\}$$

The energy E_z of a region is computed as:

$$E_z = \sum_{J_z} F_{avg}(f,g).$$

The spectral equalisation of a slice $s(x,y)$, now, consists of executing following four steps:

1. the FFT, $S(f,g)$, of $s(x,y)$ is computed;
2. each amplitude $\alpha(f,g)$ of S is compared to the statistical values of F^+ and F^-. Two issues are possible:
   - if $\alpha(f,g) \in [F^-, F^+]$, then the value remains unchanged;
   - if $\alpha(f,g) \notin [F^-, F^+]$, then $\alpha(f,g)$ is made closer to F_{avg} using: $\alpha(f,g) := (\alpha(f,g)+F_{avg})/2$. This is repeated until $\alpha(f,g) \in [F^-, F^+]$. In practice, we limited the number of iterations to about 50 in order to avoid spending too much time in equalising individual amplitudes. Our experiments have shown that higher values do not yield better results. This is due to the fact that the slice process represents only a part of the whole resolution procedure. In fact, every slice is visited more than once (see section 5);
3. on the previously defined regions J_z, we compute the corresponding energies E'_z of the slice, where $E'_z = \sum_{J_z} \alpha(f,g)$. Then, we equalise the frequencies of J_z in order to recover the same global energy amount, i.e. $\alpha(f,g) := E/E'\alpha(f,g)$;
4. the FFT^{-1} of the modified $S(f,g)$ is computed in order to get a new modified slice.

A normalised FFT permits to maintain its amplitudes lower than one, but its inverse one is limited only by N^2. Therefore, not all new colour values of the slice are between -1 and 1 (if all colour components have been previously normalised between -1 and 1). Consequently, we additionally apply a scale modification to the slice by simple dilatation, i.e. $s(x,y) = -1 + 2(s(x,y) - s_{min}(x,y))/(s_{max}(x,y) - s_{min}(x,y))$.

4.2. Spatial matching

The histogram of a picture (monochromatic) is a function defined on a discrete set and represents the relative amounts of the different hues in the picture. For a picture with n pixels, we divide the interval $T = [-1,1]$ of hues into 256 intervals of equal length $T_k = [-1 + k/128, -1 + (k+1)/128[$, where

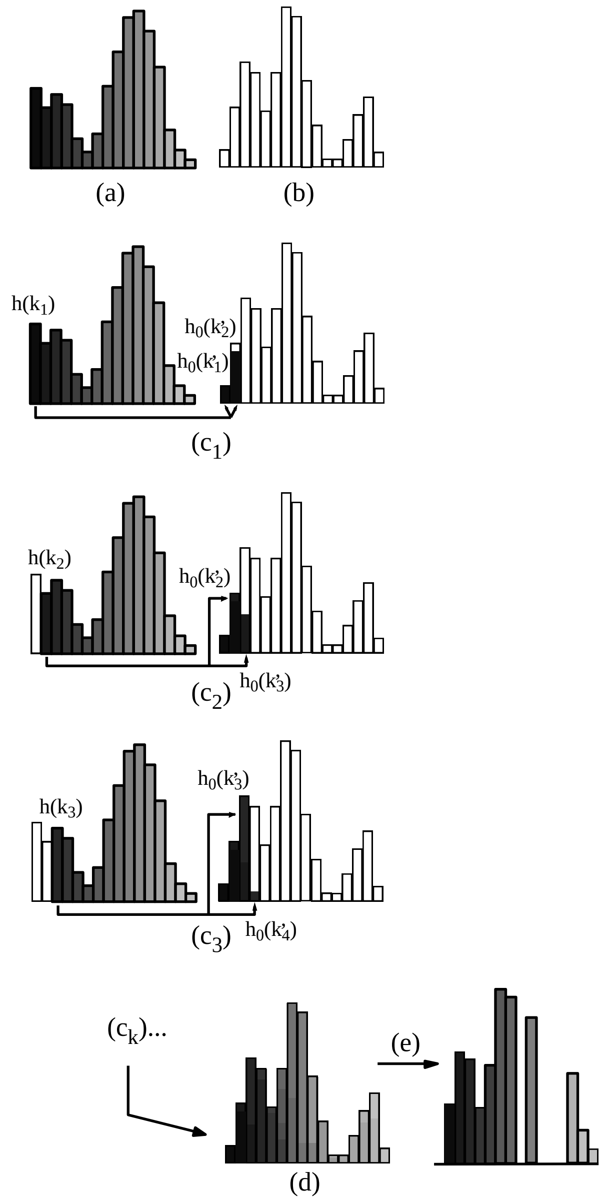

Figure 3: *Spatial-equalisation.*

$0 \leq k \leq 255$. The number $h(k)$ of hues is given by the number of pixels of T_k divided by n.

Before processing the slices, we compute a model histogram obtained by considering the digitised texture picture (the model corresponding to the direction of the slice). The major difficulty of the spatial matching process lies in the fact that we cannot dissociate pixels that have the same hue. One may only change the colour of the pixels of the entire set. Figure 3 depicts the principles of the algorithm that will be used to equalise the spatial domain.

On this figure, (a) represents the histogram h of the slice and (b) the histogram h_0 of the model. The principle consists of progressively filling the boxes of $h_0(k)$ of (b) with the contents of the boxes of $h(k)$ of (a). In intermediate steps (c_k), the entire contents of the zones of $h(k)$ are taken and distributed into the histogram (b). If a box of (a) cannot hold the entire content of the box of $h_0(k')$, the latter is segmented and the remaining part distributed as previously into $h_0(k+1)$, until $h(k)$ is completely worked out. The entire filling is guaranteed because of the normalisation of the histograms, that infers that $\sum h(k) = \sum h_0(k) = 1$.

Once the histogram (b) is entirely filled (step (d)), we need to reconstitute the boxes of $h(k)$ of (a) by grouping them again and by assigning to each of them a new index $k''(k)$, that will be computed as an average, weighted by the proportion of boxes into which the $h(k)$ have been distributed. The new slice is obtained as all pixel colours of I_k are modified using the median colour of $I_{k''(k)}$. The new slice histogram is given by (e).

5. Relaxation process

Considering figure 4, solid textures are generated by starting with a white noise block (a) and then processing all slices successively. In most cases, the resulting block (e) has yet a close visual resemblance with respect to the models. However, the procedure must be reiterated before converging on a stable solution, i.e. reusing (e) instead of (a). Experimentally, we found that 10 iterations provide good visual results for the worst cases, but in most ones such as wood grain, two or three are usually enough. We note that this procedure is different from the one we used in [5], since in this case, we do not consider the blocks and slices independently, but rather sequentially. This results in a much faster convergence than before. Indeed, with our previous method, from 50 to up to hundreds of iterations were necessary, and in some cases, still without reliable convergence.

Colour plate 1 illustrates different blocks that we

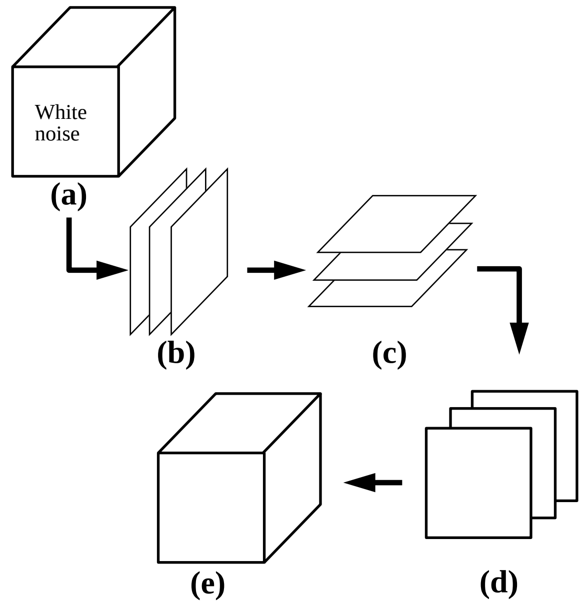

Figure 4: *Generating the solid textures.*

obtained using two or three digitised models. The blocks have a size of 128^3, and required in the worst case 10 iterations with three models and in the best case two iterations with two models. We note that different solid texture are obtained whenever the user starts with a different initial noise block. This allows a user to compute different 3D samples of a same texture.

We also note that the orientation of the model dramatically influences the result, especially if the textures are anisotropic and only two models are supplied. Figure 5 illustrates an example of wood grain. This example shows that the method works well, since in both cases, although different, the results match our expectation. The generation is a reliable process, unless the supplied models are not coherent. Still, texture mixtures may be computed if different models of different natural textures are used for the different directions.

Finally, color plate 2 illustrates a complete scene. All textures of this scene are solid textures obtained using this method. The 2D models are shown directly in the figure.

6. Limitations and discussion

In section 2, we made two restrictive assumptions. The first one is that the texture fields are stationary and contain no precise macro-structures inside (as opposed to bombing textures [17], plan tessellation, etc.). The

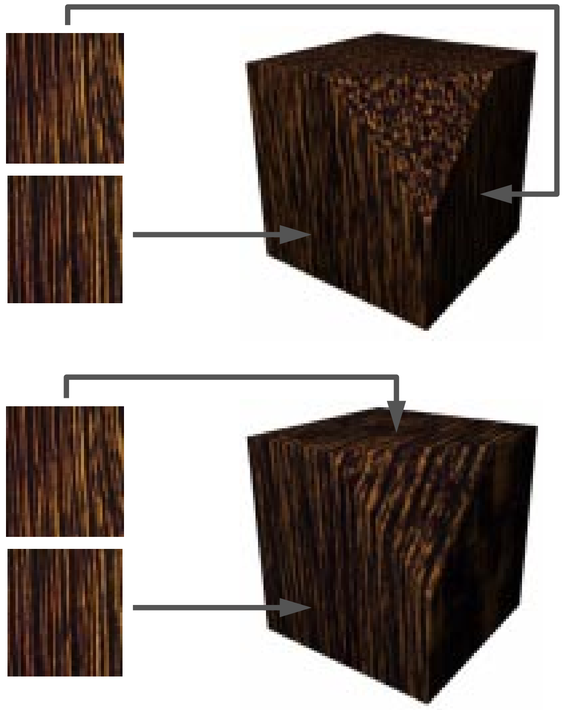

Figure 5: *Using different orientations results in different obtained solid textures.*

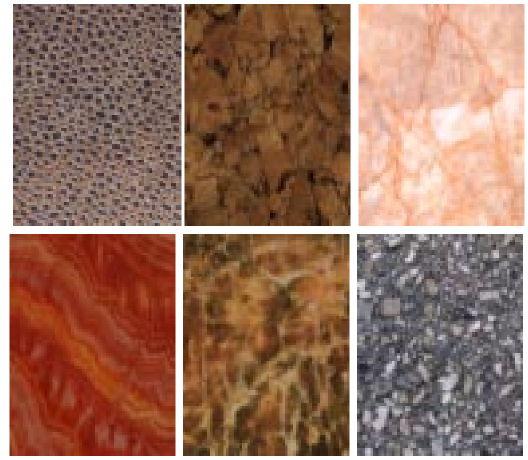

Figure 6: *Examples of structured models that cannot be processed with this method.*

method fails for example when supplying one of the models illustrated in figure 6. The reason is closely related to the fact that such textures contain important edges or are characterised by higher order statistics, that the frequency domain and histogram cannot account for. Other conceivable slice processes that gather this information more efficiently, may however, fail when applying the relaxation process. The slices should only be smoothly coerced to match the model's visual resemblance (which is difficult with edges and higher order statistics). In fact, making each individual slice close to the corresponding model is not sufficient for generating a solid block, since slices may become "independent". As a result, the 3D context may become lost. The method we use permits a "smooth" coercing and therefore works well. Macro-structured textures raise a particular problem. In fact, we first need to "guess" their actual 3D aspects before being able to build the solid textures. But reconstructing 3D shapes from only several views is a difficult problem.

The second restriction is related to the assumption of visual invariance along the axes. A priori, this seems to be an important restriction. However, it is necessary for the relaxation technique that we use. The problem of solid texture synthesis using 2D models is actually an extremely difficult problem. Consider for example a solid texture defined by following procedural model: $t(x,y,z) = \sin(\sqrt{x^2+y^2+z^2})$. The colour periodically changes as we move away from the origin. Parallel slices of this texture will not have the same appearance and it is very difficult to recover its global aspect using only a few orthogonal 2D models. But because of the impossibility to get complete 3D samples, we can only use 2D models. For this reason, we need to go a step further in analysis, and add methods in order to "guess" what is "missing", though the adequate filling of missing data is never simple. The assumption of visual invariance is a good way to do this in many cases.

7. Conclusion

2D texture analysis for automatic 3D synthesis is a very recent and difficult problem in Computer Graphics. There exists at present only a few studies that deal with this topic. Since solid textures have some obvious advantages, they are widely used in the field of realistic rendering. It is important to provide methods that allow users an easy synthesis of new solid textures. Analysis is an approach that can plainly satisfy this request, and therefore has known some recent success.

In this paper, we have presented an efficient method for automatically generating solid textures from several orthogonal 2D views using an assumption of visual invariance and stationarity. The method deals with a wide range of solid textures, including the difficult case of anisotropic textures, that previous techniques could not process correctly. It represents, in fact, the first reliable method that permits the synthesis of a great number of solid textures. In addition, it allows an efficient control of the entire aspect of the textures.

This is well shown by the figure 5 and colour plate 1, where the obtained textures are very similar to the natural texture models. beyond our expectation. Still, the method is limited.

In section 6, we have discussed some limitations due to the use of 2D models. A universal solution to the problem of automatic solid texture synthesis will be hard to find. A more appropriate approach will surely consist of separating the different classes of textures. For example, some current research consists of efficiently tackling "macro-structured" textures. A pre-process, based on shape analysis techniques frequently used to recover 3D shapes from 2D views, should help, for example, to process the particular case of "bombing" textures [17].

References

1. Peachey D., *Solid Texturing on complex Surfaces*, Computer Graphics **19**(3), July 1985, pp. 279–286.
2. Perlin K., *An Image Synthesizer*, Computer Graphics **19**3, July 1985, pp. 287–296.
3. Ebert D.S., Musgrave F.K., Peachey D., Perlin K. and Worley S., *Texturing and Modeling: A Procedural Approach*, Academic Press, 1994.
4. Ghazanfarpour D. and Dischler J.M., *Spectral Analysis for Automatic 3-D Texture Generation*, Computers and Graphics **19**(3), 1995, pp. 413–422.
5. Ghazanfarpour D. and Dischler J.M., *Generation of 3D-Textures Using Multiple 2D Models Analysis*, Computer Graphics Forum **15**, (Eurographics'96), 1996, pp. 311–323.
6. Heeger D.J. and Bergen J.R. *Pyramid-Based Texture Analysis/Synthesis*, Computer Graphics **29**, August 1995, pp. 229–238.
7. Bergen J.R. and Adelson E.H., *Early vision and texture perception*, Nature, **333**, 1988, pp. 363–364.
8. Julesz B., *Textons, the Elements of Texture Perception, and their Interactions*, Nature **290**(London), 1981, pp. 91–97.
9. Dyer C.R. and Rosenfeld A., *Fourier Texture Features: Suppression of Aperture Effects*, IEEE Transactions on Systems, Man, and Cybernetics, October 1976, pp 703–705.
10. Matsuyama T., Miura S. and Nagao M., *Structural Analysis of Natural Textures by Fourier Transformation*, Computer Vision, Graphics and Image Processing **24**, 1983, pp 347–362.
11. Lowitz G.E., *Can a local Histogram really map Texture Information?*, Pattern Recognition **16**(2), 1983, pp. 141–147.
12. Clark M. and Bovik A.C., *Experiments in Segmenting Texton Patterns Using Localized Spatial Filters*, Pattern Recognition **22**(6), 1989, pp. 707–717.
13. Dunn D., Higgins W.E. and Wakeley J., *Texture Segmentation Using 2-D Gabor Elementary Functions*, IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(2), February 1994, pp. 130–149.
14. Turner M.R., *Texture Discrimination by Gabor Functions*, Biological Cybernetics **55**, 1986, pp 71–82.
15. Francos J.M. and Zvi Meiri A., *A Unified Structural Stochastic Model for Texture Analysis and Synthesis*, in proceedings of 9th Int. Conf. Patt. Recognition, Rome, 1988, pp. 41–45.
16. De Bonet J.S., *Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images*, Computer Graphics **31**, 1997, pp. 361–368.
17. Schachter B.J. and Ahuja N., *Random pattern generation processes*, Computer Graphics and Image Processing **10**, 1979, pp. 95–114.

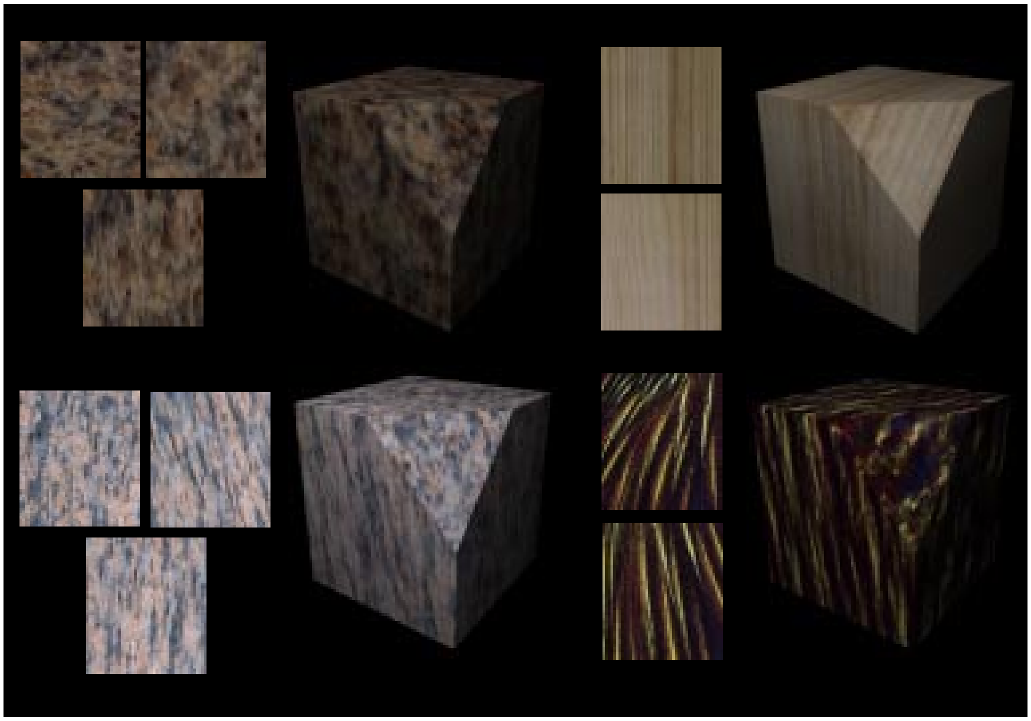

Figure 7: Colour plate 1. *Examples of anisotropic solid textures generated with the relaxation technique using different orthogonal models.*

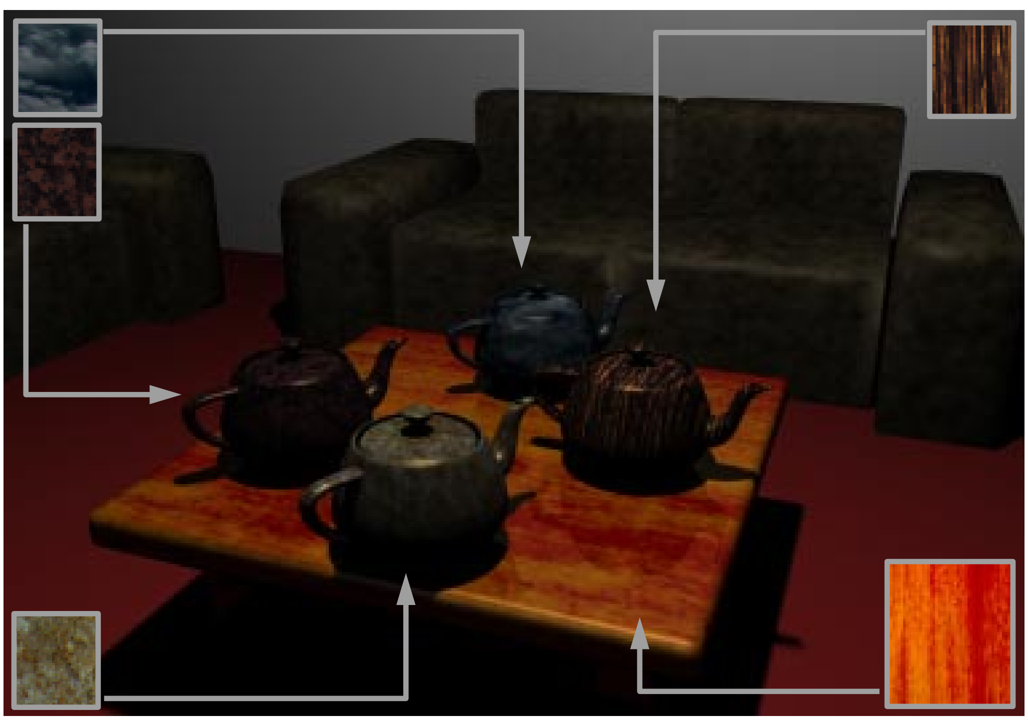

Figure 8: Colour plate 2. *A scene showing different solid textures and the corresponding models.*