Maximum-likelihood based synthesis of volumetric textures from a 2D sample

Radu-Dragos Urs, Jean-Pierre da Costa, Christian Germain

▶ **To cite this version:**

Radu-Dragos Urs, Jean-Pierre da Costa, Christian Germain. Maximum-likelihood based synthesis of volumetric textures from a 2D sample. IEEE Transactions on Image Processing, 2014, 23 (4), pp.1820-1830. hal-00978375

HAL Id: hal-00978375
https://hal.science/hal-00978375v1

Submitted on 14 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Maximum-likelihood based synthesis of volumetric textures from a 2D sample

R.-D. Urs, J.-P. Da Costa, C. Germain

***Abstract* — We propose a genuine 3D texture synthesis algorithm based on a probabilistic 2D Markov Random Field conceptualization, capable of capturing the visual characteristics of a texture into a unique statistical texture model. We intend to reproduce, in the volumetric texture, the interactions between pixels learned in an input 2D image. The learning is done by non-parametric Parzen-windowing. Optimization is handled voxel by voxel by a relaxation algorithm, aiming at maximizing the likelihood of each voxel in terms of its local conditional probability function. Variants are proposed regarding the relaxation algorithm and the heuristic strategies used for the simultaneous handling of the orthogonal slices containing the voxel. The procedures are materialized on various textures through a comparative study and a sensitivity analysis, highlighting the variants strengths and weaknesses. Finally, the probabilistic model is compared objectively with a non-parametric neighborhood-search based algorithm.**

***Index Terms* — volumetric texture, non-parametric synthesis, multiresolution, Markov Random Field, conditional probability, iterated conditional modes, stochastic relaxation**

This work is funded by the French National Research Agency and by Aerospace Valley, World Competitiveness Cluster in Aeronautics, Space and Embedded Systems of Midi-Pyrénées and Aquitaine.

Radu-Dragos Urs is with the Signal and Image Processing Group within the IMS Laboratory (*Laboratoire de l'Intégration du Matériau au Système*), CNRS UMR 5218, F33405 Talence, within University of Bordeaux, France, Tel : (+33)540006185, e-mail: radu.urs@ims-bordeaux.fr.

Jean-Pierre Da Costa and Christian Germain are with the Signal and Image Processing Group of the IMS Laboratory and affiliated to Bordeaux Sciences Agro, University of Bordeaux, F33175 Gradignan, France, e-mails: jean-pierre.dacosta@ims-bordeaux.fr, christian.germain@ims-bordeaux.fr.

I. Introduction

The effectiveness of image synthesis techniques to model and reproduce natural textures (wood, rock, sand, fabric etc.) does not have to be anymore demonstrated. The field of texture synthesis has been particularly dynamic, with notable applications in image extrapolation, editing, compression or mapping but has also been extended to other areas such as video completion, merging and animations, or description of the geometry of a surface. Despite the numerous successful 2D texture synthesis studies, in terms of quality and efficiency, the solid texture synthesis receives relatively less attention. The scarcity of related papers is attributed to the much higher complexity involved in solid texture synthesis. 3D textures are mainly used for texturing volumetric objects trying to increase the realism of 3D scenarios, but they can also be observed in 3D vision when exploring material structure, seismic data or medical images. Unlike 2D image samples, capturable with any camera, 3D images are more delicate to obtain. In material science and engineering, exploring a material 3D structure is essential to understand and predict its physical properties and behavior. There are different 3D imaging techniques (ultrasound imaging, tomography, magnetic resonance imaging, coherent diffraction imaging) allowing to obtain 3D views of a material at different resolutions, but their use is not always appropriate due to the high costs they involve and for various technical reasons (e.g. sample preparation, high resolution intricacy). 3D image synthesis may appear as an interesting alternative in such cases. Specific techniques are to be considered, using at least one 2D image of the material as input. 3D image data can be obtained from one 2D image by constraining the targeted solid block to match the 2D sample characteristics.

In the following, we focus on 3D texture synthesis from a 2D sample. After a review of related techniques, we propose a genuine non-parametric algorithm for the synthesis of textures considered as Markov Random Fields, based on the patch likelihood maximization in the sense of the local conditional probability function. We show and discuss synthesis results while carrying out a comparison with a reference algorithm. Finally, some conclusions and prospects are provided.

II. Related solid texture synthesis approaches

Full 3D (i.e. 3D input, 3D output) texture synthesis versions can be easily derived for most of existing 2D approaches, provided that a 3D sample is available as an exemplar. On the contrary, the synthesis of volumetric textures based on a 2D sample is more awkward, distinguishing three main directions for creating volumetric textures, as follows.

A. Procedural approaches

These are the first 3D synthesis techniques ever attempted, reproducing realistic textures by perturbing noise functions to create pseudo-random patterns [2-4]. These can produce solid textures containing patterns of marble, sand, clouds, fire or water. Another efficient solid synthesis algorithm simulating the growth of a material from a single 2D grow-able texture pattern was presented in [5]. Texture evolution is based on texture growing and turbulence. However it is quite difficult to analytically

describe texture appearance by procedural methods leading to algorithms hard to control and optimize.

B. Internal solid texturing from 2D cross sections

It consists in synthesizing the internal structure of a 3D model from 2D snapshots of a few cross-sections of a real object instead of sampling directly from a complete 3D volumetric representation [6,7]. While in [6] the user is asked to provide a direction to orient textures inside the volume, [7] calls for the user to place the cross sections onto the 3D model and performs the cross-sections synthesis by morphing. Their 3D interactivity is very appealing even for a non-expert user, but these methods show discrepancy among cross-sections, high computational cost and morphing limitations. They work well with isotropic, layered and oriented textures, but fail on textures containing discontinuous elements. Similar work was accomplished in [8] by filling a solid object with overlapping solid textures, by extending a 2D patch-pasting approach [9].

C. Exemplar-based texture synthesis methods

These are the most frequently employed aiming to fill up a volume with the patterns seen in the exemplar, by inventing 3D information from 2D input data, as discussed in [1]. The first parametric method involved multi-scale statistical feature matching operations, to reproduce the global statistics of the 2D exemplar in the volume. 3D textures are successfully generated by matching the histogram of the volumetric data with that of the input sample at different resolutions [10].

Work in this direction was extended following the idea that an image can be relevantly decomposed using a bank of spatial filters, into a set of sub-bands, each sub-band revealing information about the presence of primitives of specific orientation and scale in the exemplar [11,12]. The statistical modeling of these sub-bands was extended to the 2^{nd} order using spatial autocorrelations. These methods work well on homogenous and stochastic textures, but the quality of the synthetic results deteriorates in general for structured textures.

Other attempts are made in order to match the spectral characteristics of the 2D exemplar [13,14]. Based on a Fourier spectrum analysis of the model, a basis and a noise function are obtained and then a procedural algorithm is used to get the 3D texture as in [2]. Later work combined spectrum analysis with histogram matching and used orthogonal 2D views in order to synthesize anisotropic solid textures [15]. Despite the limited range of handled textures whose appearance is well captured by global parameters, it is able to generate structural solid textures such as wood and marble.

To overcome the restricted applicability of the synthesis methods, non-parametric approaches were later proposed [16-19]. These are essentially based upon the assumption that textures are Markov Random Fields. The solid texture is generated systematically one voxel/patch at a time maintaining the coherence of the local texture with its vicinity. The global schema consists in copying pixels from the exemplar into the output texture, making sure that the output voxel 2D neighborhoods – taken on up to three orthogonal 2D slices containing the output voxel – are similar to the neighborhoods of selected input pixels [16].

Improvements were brought in [17] that combined global texture optimization [19] with color histogram matching as in [10]. To upgrade even more the quality of the results, [18] proposed to integrate into the texture optimization process two new kinds of matching histograms (position and index histograms). An objective comparison of these fixed-neighborhood search based algorithms is carried out in [20] in the case of specific laminar textures. The algorithms occasionally fail to reproduce the structure or tend to synthesize textures with verbatim repetitions. The non-parametric pixel by pixel synthesis methods find applications even in modeling real human body systems (e.g. for creating realistic 3D organic tissues [21]).

To take into account long term dependences observed in real textures while preserving a reasonable computational complexity, most authors propose multi-scale implementation schemes accelerated with efficient searching algorithms (tree structured vector quantization [22], k-coherence tree [23], discrete solver [24]) or dimensionality reduction [25].

In [26], a mathematical framework is proposed for generating solid textures by sampling the Grey Level Aura Matrices of the input samples in multiple view directions. It characterizes the co-occurrence probability distributions of grey levels at all possible displacement configurations. The results are impressive, but for handling with colors, color channels must be uncorrelated; while in most textures the channels are strongly correlated [17] and independently synthesizing the uncorrelated channels leads to visual artifacts.

Solutions based on stereological techniques were attempted by modeling the shape and the spatial distribution of the particles observed on 2D images of binding materials [27,28]. To synthesize solid textures of particles, a first proposition was to operate in the spatial domain in three steps: detect and extract the particles structures from the exemplar, model the extracted shapes using generalized cylinders and distribute the 3D reconstructed shapes inside a volume [27]. [29] addressed the challenges of creating solid representations of aggregate materials; it accurately estimated the 3D particle distribution from a 2D image using particle shape models.

A technique that enables 2D synthesis of textures visually indistinguishable from their models is the non-parametric Markov Random Field (NP-MRF) texture modeling method [30]. This method mathematically captures the visual characteristics of a texture into a unique statistical texture model [31] that describes the interactions between pixels values, based on the conditioned neighborhood concept. The relevance of the model is validated by synthesizing 2D textures. Each output pixel is given the most probable value, considering only the neighboring pixels. This model is capable of synthesizing complex 2D textures ranging from the stochastic to the well-structured ones. However, this approach, though efficient and theoretically legitimate, was initially proposed in 2D and has never been extended in 3D.

In this paper, we propose a genuine 2D/3D synthesis method, doing explicitly what the methods based on strict neighborhood search [16-18] tried to accomplish indirectly, i.e. finding for each voxel the best value with regard to its neighborhood. Our approach extends in 3D the MRF-modeling concept first proposed in 2D in [30]. We stick to the idea of having the input/output

configurations identically distributed, in terms of the local conditional probability density function. The texture is generated voxel by voxel by maximizing a heuristic function of the voxel likelihoods expressed in several orthogonal 2D sections.

In the following we deal with some required MRF theory used to describe the non-parametric synthesis process. Then we present our novel 3D operating method and its several variants. An experimental study is carried out allowing a visual evaluation of the method and its objective quantitative comparison with a state of the art non-parametric approach [17]. After a short discussion about 2D/3D inference, some conclusions and prospects are finally given.

III. NP-MRF TEXTURE MODEL

A. MRF Model: concept and theory

MRFs have been studied extensively for solving various image analysis problems [30-36], like image restoration and segmentation, edge detection, texture analysis and synthesis. The MRF principle can be seen as a sites-and-labels concept - the solution to a problem is a set of labels assigned to a set of sites. A label is an event that may happen to a site and the labeling operation is to assign a label to each site. Translating this into MRF image modeling, one retrieves that an image of size $w \times h$ corresponds to a lattice (i.e. set of sites) $S = \{s_1, s_2 \dots s_{w \times h}\}$, so that each pixel in the image is a site s on the related lattice and its value x_s belongs to the label set $\Lambda = \{0,1 \dots L-1\}$, with L being the number of possible values. The labeling, also called a configuration, relates to the image itself, being seen as a mapping function from S to Λ:

$$f: S \to \Lambda, \quad f(s) = x \tag{1}$$

To develop the MRF theory, the spatial dependencies between sites (image pixels) have to be defined. This is done by choosing the neighborhood system, i.e. the set of all neighborhoods $N_s = (i,j) \in S$, the neighborhood of a site *s* being the set of neighboring sites [30,33].

The main property of a MRF can be stated as follows. *The random variable X_s describing the intensity value at any site s can take any value $x_s \in \Lambda$, but the probability that $X_s = x_s$ depends only on the values at neighbouring sites r* [32,33]:

$$\mathcal{L}(\mathrm{s}) = \mathrm{P}(\mathrm{X_s} = \mathrm{x_s} | \mathrm{X_r} = \mathrm{x_r}, \mathrm{r} \neq \mathrm{s}) = \mathrm{P}(\mathrm{x_s} | \mathrm{x_r}, \mathrm{r} \in \mathrm{N_s}) \tag{2}$$

This conditional probability (i.e. markovianity condition) is termed as *local conditional probability density function* (LCPDF). MRFs can be specified in terms of their conditional probabilities and, inversely, MRFs can be of help in deducing the joint probability distribution from the associated local conditional probabilities. The joint distribution defines the probability for a particular labeling realization. A valid joint distribution is in fact uniquely defined by its LCPDF, by applying the "Markov - Gibbs equivalence theorem" [33] or the "Hammersley - Clifford theorem" [34], establishing the equivalence between the local property (markovianity) and the global property (Gibbs distribution of a Gibbs Random Field).

B. Local Conditional Probability Density Function

The NP-MRF model estimates the LCPDF from a multi-dimensional histogram of the neighborhood over a homogeneous (i.e. stationary) textured image [30]. Each histogram dimension represents a site from the neighborhood system considered in (2), the model's statistical order being the number of dimensions [30]. A common non-parametric estimator, that spreads each sample data into a smooth multi-dimensional histogram over a larger area, is the Parzen-window density estimator [37]. It associates each point in the histogram with a multi-dimensional density.

Let $z_p = Col\left[y_p, y_q, q \in N_p\right]$ be a column vector of size d, containing the pixel value y_p and all the values within N_p, from the input sample Y. A given configuration in the output texture X, on which the Parzen-window estimator is to be applied, is referred to as $z_{x_s} = Col[x_s, x_r, r \in N_s]$. The NP-MRF model gets to be uniquely defined by the non-parametric estimation $\hat{\mathcal{L}}(s)$ of the LCPDF $\mathcal{L}(s)$ [30]:

$$\hat{\mathcal{L}}(\mathrm{s}) = \hat{\mathrm{P}}(\mathrm{x_s}|\mathrm{x_r}, \mathrm{r} \in \mathrm{N_s}) = \frac{\hat{\mathrm{f}}(\mathrm{z_{x_S}}=\mathrm{Col}[\mathrm{x_S},\mathrm{x_r},\mathrm{r}\in\mathrm{N_S}])}{\sum_{\lambda_S\in\Lambda}\hat{\mathrm{f}}(\mathrm{z_{\lambda_S}}=\mathrm{Col}[\lambda_S,\mathrm{x_r},\mathrm{r}\in\mathrm{N_S}])} \tag{3}$$

where $\hat{\mathrm{f}}$ is the Parzen window estimated frequency:

$$\hat{\mathrm{f}}(z) = \frac{1}{n \cdot h^d}\sum_{p\in S_y, N_p \subset S_y} K\left\{\frac{1}{h}\left(z - z_p\right)\right\} \tag{4}$$

n is the number of possible neighborhoods N_p in Y, i.e. the number of sites $p \in S_y$ for which $N_p \subset S_y$. The shape of the smoothing is defined by the kernel function K, chosen to be the standard multi-dimensional Gaussian density function:

$$\mathrm{K(z)} = \frac{1}{(2\pi)^{\mathrm{d}/2}}\exp\left(-\frac{1}{2}\mathrm{z}^{\mathrm{T}}\cdot\mathrm{z}\right) \tag{5}$$

where $(\,)^T$ indicates the matrix transpose.

The size of K is modified by the window parameter h in (4). [37] provides an optimal value for h, assuring that the estimation is close to the true density function:

$$\mathrm{h_{opt}} = \sigma\left[\frac{4}{\mathrm{n(d+2)}}\right]^{1/(\mathrm{d}+4)} \tag{6}$$

where σ^2 is the marginal variance of the sample texture.

Exploiting the LCPDF estimator in (4), a value x_s of a pixel at site s depends only on the values x_r at sites neighboring s:

$$\hat{\mathcal{L}}(\mathrm{s}) = \frac{\sum_{\mathrm{p}\in\mathrm{S_y},\mathrm{N_p}\subset\mathrm{S_y}}\exp\left[-\frac{1}{2\mathrm{h}^2}\left(\mathrm{z_{x_S}}-\mathrm{z_p}\right)^{\mathrm{T}}\left(\mathrm{z_{x_S}}-\mathrm{z_p}\right)\right]}{\sum_{\lambda_S\in\Lambda}\sum_{\mathrm{p}\in\mathrm{S_y},\mathrm{N_p}\subset\mathrm{S_y}}\exp\left[-\frac{1}{2\mathrm{h}^2}\left(\mathrm{z_{\lambda_S}}-\mathrm{z_p}\right)^{\mathrm{T}}\left(\mathrm{z_{\lambda_S}}-\mathrm{z_p}\right)\right]} \tag{7}$$

C. Texture synthesis via relaxation

Texture synthesis from an MRF model is based upon a relaxation algorithm [33] guided by the LCPDF. More often used and fairly more known are the stochastic relaxation algorithms like the Metropolis algorithm [38] and the Gibbs sampler [33]. A deterministic relaxation was introduced in [31], called Iterated Conditional Modes (ICM), suggesting that it returns the mode of

the LCPDF. The synthesis starts with an image and iteratively updates the pixels with respect to the LCPDF, obtaining a sequence of images $\{x(0), x(1), \dots, x(n)\}$ in order to reach the joint distribution of $x(n)$ [30].

D. Multi-scale implementation

The multi-scale implementation aims both at capturing the textural patterns at different scales (i.e. the local interactions between pixel values) and at speeding up the relaxation process. It helps the relaxation algorithm converge to an image closer to the global maximum of the joint distribution [39]. Having an image at a local maximum of the LCPDF is in general enough to obtain a satisfactory synthetic texture.

The multi-scale grid representation is employed as in [31]. It consists in obtaining the higher grid levels *l>0*, from the image at level *l=0* by using pixel decimation. High frequency texture details are important at high resolutions, while lower resolutions synthesis behaves better on low frequency features. The relaxation starts at the highest level *l=L* and continues from one level to another until *l=0*; once a level *l* was synthesized (i.e. relaxed or reached a maximum), to pass to the next level *l+1*, all the pixels from the smaller level *l* are copied in their corresponding positions at the higher level *l+1* and then the synthesis carries on at the upgraded level.

E. Pixel confidence

The original 2D synthesis process [30] based on ICM relaxation incorporates a pixel temperature function to serve as a degree of confidence relative to the pixel value. The pixel confidence is associated with the probability that a value x_s represents the correct pixel value for the site *s*. Each pixel is given a certain temperature t_s between 0 and 1, where 0 means complete confidence. The synthesis starts at a high global temperature (i.e. no confidence) and the process is considered to be finished when equilibrium is reached – temperature 0 for all pixels. Full pixel confidence happens when x_s is sampled from an LCPDF at equilibrium, or when the LCPDF is completely conditioned by its neighboring pixel values. Once a pixel is relaxed, its temperature t_s is updated by relying only on the temperature of the neighboring pixels $t_r, r \epsilon N_s$:

$$\mathrm{t_s} = \max\{0, (\xi + \textstyle\sum_{\mathrm{r}\in\mathrm{N_s}} \mathrm{t_r})/|\mathrm{N_s}|\}, \xi < 0 \tag{8}$$

ξ is used to control the rate of the cooling schedule, that is the rate at which a pixel is cooled based on its neighbors temperatures. At high temperatures the synthesized textures are not spatially correlated, but as the temperatures decrease, the correlations induced by the LCPDF become stronger and the cooling rate should be slowed down to capture correctly the image characteristics (i.e. spatial correlations). In the ICM based MRF model, the pixel temperature function is used to condition the LCPDF: if $t_s = 0$ the LCPDF should be strong, while for $t_s = 1$ it should be weaker. To ensure this, the pixel confidence is integrated in the form of $(z_{x_s} - z_p)$ in (7):

$$(\mathrm{z_{x_s}} - \mathrm{z_p}) = \mathrm{Col}[\mathrm{x_s} - \mathrm{y_p}, (\mathrm{x_r} - \mathrm{y_r}) \cdot (1 - \mathrm{t_r}), \mathrm{r} \in \mathrm{N_s}] \tag{9}$$

This cooling process suggests that the relaxation algorithm is a kind of annealing schedule [33]. However, it is important to note that the temperature function is used in [30] as a way to manage multi-resolution and that this relaxation algorithm is definitely deterministic, not stochastic.

IV. 3D EXTENSION OF THE NP-MRF SYNTHESIS ALGORITHM

A. 2D-3D generic algorithm

Our 3D texture synthesis algorithm (illustrated in *Figure 1*) is an extension of the NP-MRF algorithm. It requires only a 2D texture sample as an input, even if variants could be easily derived with 2 or 3 input samples. The targeted output volume is filled up with white noise. Both are decomposed at different scales and defined as lattice systems. Two versions of the algorithm are provided. The first one is based on deterministic ICM relaxation as in [30], while the second is based on stochastic relaxation [33].

With ICM relaxation, every 3D site, i.e. every voxel, is given a temperature of 1. The synthesis starts at the highest grid level and proceeds by randomly choosing sites from the lattice. If the temperature of a site is positive, the value of that site is questioned. The decision is made by finding the value, within the available configuration, that maximizes the LCPDF of that site. After a site is treated, its temperature is updated. If the temperature reaches 0, the corresponding pixel is considered to have a correct value and LCPDF estimation is no longer computed. Once the lattice is fully scanned, the global temperature (i.e. sum of all temperatures) is computed, the scan ending when the global temperature becomes 0.

For the stochastic relaxation, the algorithm is a simulated annealing [33]. It requires a single temperature value t_{in} used to determine the number of random draws within a scale. A new randomly selected value is accepted as update value for a site s if its LCPDF is higher than the LCPDF associated to the former value x_s. If not, this new value could also be accepted with a certain probability, related to the LCPDF ratio and to the overall temperature t_{in}. After each random draw, the temperature is decreased so that as the relaxation advances new changes are less accepted.

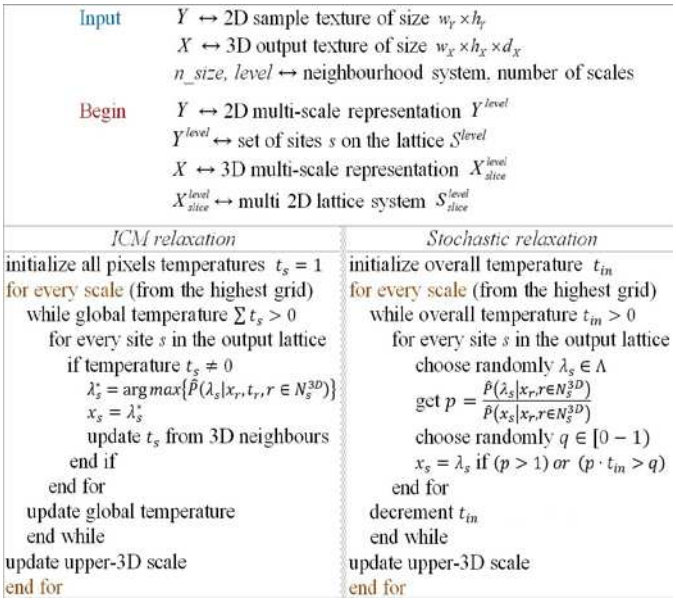

Figure 1 – *Algorithm: volumetric texture synthesis using NP-MRF model.*

B. Scale handling

The decomposition of the texture from a low scale to a higher scale is done by simple decimation as in [30]. The contrary top-down operation is systematically done after each synthesized scale by copying each high scale pixel value into one or eight lower scale pixels (1/8 or 8/8 up-sampling). 1/8 up-sampling is used for ICM relaxation to guarantee multi-resolution coherence while allowing some randomness in the initialization since the seven other pixels are initialized at random. As stochastic relaxation allows randomization by nature, 8/8 up-sampling is sufficient when using simulated annealing.

In the case of the deterministic relaxation based on the pixel confidence function, the temperatures of the high scale voxels can be copied to the next lower scale only after the high scale is synthesized. This means that the high scale voxels have full confidence ($t_s = 0$) and they are propagated onto the next lower scale keeping the same temperature. These voxels are thus used as fixed voxels. This allows a faster synthesis, but not necessary a better one. An erroneous voxel can propagate from an intermediary level to the target texture. We prefer to use the pixels from a higher scale to initialize the next lower scale, but to reconsider them during synthesis [40]. The value of a voxel from a previous level is changeable but its temperature remains fixed, giving it a full confidence for its neighbors. This pair pixel re-synthesizing phase increases slightly the computational cost but provides noise-free results.

C. Why isn't a full-3D process relevant?

The difficulties encountered at this point put forward the question on how to infer 3D information when only 2D information is available. The issue is to find a good strategy capable of working out the LCPDF of a site on the 3D grid by means of information available on the sample's 2D lattice. The ideal solution would be to figure out the full-3D LCPDF associated to a

site *s* in 3D, and subsequently to retrieve its associated value λ_s^* so that the LCPDF would be completely conditional on its full-3D neighborhood N_s^{3D}:

$$\lambda_s^* = \arg\max\{\mathcal{L}_{3D}(\lambda_s)\} = \arg\max\{\widehat{P}(\lambda_s|x_r, r \in N_s^{3D})\} \quad (10)$$

However, since only 2D information is available for learning, it is not possible to estimate the conditional probability of such 3D configurations to assess the interactions inside 3D neighborhoods. This observation led us to propose heuristic approaches (in *Figure 3*) to overcome the problem.

D. Heuristic approaches

Our extension infers the output block by treating it in a multi-2D way, as an arrangement of manifold 2D sheets. It implies a multi-2D/2D synthesis that comes down to splitting the 3D neighborhood into orthogonal 2D neighborhoods. The outcome is in reducing the unknown full-3D LCPDF estimation to a compound of computable 2D estimations.

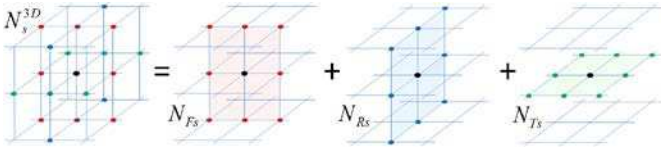

Figure 2 – *Breaking a 3D neighborhood N_s^{3D} into three 2D neighborhoods related to the orthogonal views: front N_{Fs}, right N_{Rs} and top N_{Ts}.*

Figure 2 shows the breaking of a 3D lattice neighborhood N_s^{3D} into three orthogonal 2D neighborhoods (front N_{Fs}, right N_{Rs} and top neighborhood N_{Ts} related to the central site) as suggested in [16-18]:

$$N_{s=(i,j,k)}^{3D} = N_{F\,s=(i,j)} \cup N_{R\,s=(j,k)} \cup N_{T\,s=(i,k)} \quad (11)$$

This kind of conceptualization shows the need for heuristics to attain for voxel v_s the maximum likelihood (ML) value λ_s.

1) Heuristic 1: ML_H1

According to the 3D neighborhood breaking scheme one can find a suitable voxel value v_s at site *s*, as a combination of the most probable values found separately for each of its orthogonal views, as a 2D/2D search problem. The result contains several solutions and LCPDFs, one pair for each view - $(\lambda_{front}, \hat{P}_{front})$, $(\lambda_{right}, \hat{P}_{right})$ and $(\lambda_{top}, \hat{P}_{top})$, retrieved through separate maximum searches:

$$\lambda_{view}^* = argmax\{\hat{P}_{view}(\lambda_{view}|x_r, r \in N_{view\_s})\},\ \lambda_{view} \in \Lambda \quad (12)$$

It reminds of the nearest neighbor search based algorithms, where the voxel update value is computed as a combination of the solutions of each orthogonal view [16-18]. Similarly, the chosen update value can be found by combining the values proposed by each orthogonal view using as strategy:

- the one with the lowest related LCPDF, tagged as *ML_H1a*;

- the average, tagged as *ML_H1b*.

2) Heuristic 2: ML_H2

In order to avoid performing several maximum searches and dealing with several solutions for each orthogonal view, a second heuristic is proposed. It uses a function *f* of the 2D estimations associated to the orthogonal views, for instance:

$$\mathrm{f}\left(\mathcal{L}_{\mathrm{front}}, \mathcal{L}_{\mathrm{right}}, \mathcal{L}_{\mathrm{top}}\right) = \min\left(\mathcal{L}_{\mathrm{front}}, \mathcal{L}_{\mathrm{right}}, \mathcal{L}_{\mathrm{top}}\right) \tag{13}$$

$$\mathrm{f}\left(\mathcal{L}_{\mathrm{front}}, \mathcal{L}_{\mathrm{right}}, \mathcal{L}_{\mathrm{top}}\right) = \mathrm{product}\left(\mathcal{L}_{\mathrm{front}}, \mathcal{L}_{\mathrm{right}}, \mathcal{L}_{\mathrm{top}}\right) \tag{14}$$

that when maximized, allows to find the update value λ_s^* :

$$\lambda_{\mathrm{s}}^* = \mathrm{argmax}\left\{\mathrm{f}\left(\mathcal{L}_{\mathrm{front}}(\lambda_{\mathrm{s}}), \mathcal{L}_{\mathrm{right}}(\lambda_{\mathrm{s}}), \mathcal{L}_{\mathrm{top}}(\lambda_{\mathrm{s}})\right)\right\}, \lambda_{\mathrm{s}} \in \Lambda$$

$$\text{with } \mathcal{L}_{\mathrm{front}}(\lambda_{\mathrm{s}}) = \widehat{\mathrm{P}}_{\mathrm{front}}(\lambda_{\mathrm{s}} | \mathrm{x}_{\mathrm{r}}, \mathrm{r} \in \mathrm{N}_{\mathrm{Fs}}),$$

$$\mathcal{L}_{\mathrm{right}}(\lambda_{\mathrm{s}}) = \widehat{\mathrm{P}}_{\mathrm{right}}(\lambda_{\mathrm{s}} | \mathrm{x}_{\mathrm{r}}, \mathrm{r} \in \mathrm{N}_{\mathrm{Rs}}),$$

$$\mathcal{L}_{\mathrm{top}}(\lambda_{\mathrm{s}}) = \widehat{\mathrm{P}}_{\mathrm{top}}(\lambda_{\mathrm{s}} | \mathrm{x}_{\mathrm{r}}, \mathrm{r} \in \mathrm{N}_{\mathrm{Ts}}) \tag{15}$$

These heuristics are tagged as follows:

- *ML_H2a* for the expression in (13) that maximizes the smallest of the likelihoods in order to maximize them all,
- *ML_H2b* for an *f* defined as in (14) to maximize the product, searching for a compromise for all the three views.

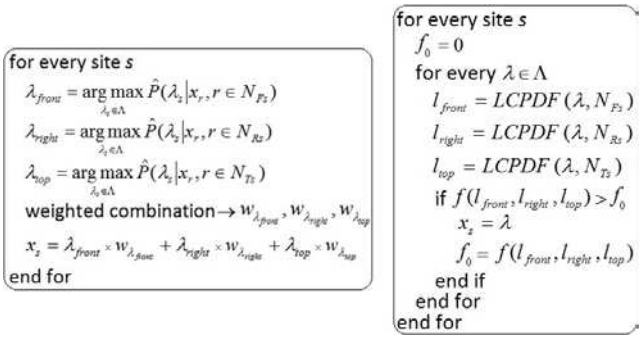

Figure 3 – *Principle of heuristic: ML_H1 (on left) and ML_H2 (on right).*

V. SYNTHESIS RESULTS

This section deals with putting in practice the maximum-likelihood based non-parametric synthesis algorithm presented in section IV. The proposed heuristics are applied to the volumetric texture synthesis starting from a single 2D texture. The goal is to analyze the algorithm's capacity to synthesize different types of grey-level and color textures, to show the potential of these methods in various domains. The synthesized result should resemble the source texture, preserving the dynamics and the structure.

In terms of synthesis quality, a critical point consists in taking into account the size of the texture patterns in determining the number of scales and the neighborhood size. Other more specific factors are for the ICM based algorithm the temperature

decreasing scheme, the synthesis ending criteria and the up-sampling procedure. For the stochastic relaxation, the number of draws plays a key role both in computational cost and in texture quality. By testing these factors influence, one can identify the relevant strategy to obtain good quality synthetic results.

Care is taken to ensure that the extrapolation of the texture in 3D makes sense in terms of anisotropy. The existence of a real 3D structure, giving rise to a 2D section similar to the example, has to be possible. The algorithms have to be parameterized so that the synthesis of such 3D structures is possible. A discussion over the number of views of the solid output block, to be constrained by the input sample, is dealt with in a 2D/3D inference dedicated section. Firstly, some results are shown using the parameters for which eye-friendly results are obtained, and next the influence of these parameters is analyzed on the synthetic solid textures.

A. Comparing the heuristics

To show the proposed algorithms capacity to synthesize various textures, the first results are obtained on a generic framework that experimentally proved to produce satisfying results: a neighborhood system composed of a full-square neighborhood of size 7×7, computed on 3 scales, synthesizing the voxels in a random way, 8/8 up-sampling strategy, re-synthesizing the pixels inherited from a higher scale while using the ICM relaxation. To assure good quality results in a reasonable time, the pixels temperatures are decreased by a step $\xi = -3$ in (8) and the relaxation is stopped after reaching a 95% decrease of the global temperature.

Figure 4 shows some volumetric results obtained by using the proposed heuristics. The input grey-level images are of size 64^2 pixels and the synthesized blocks - 64^3 voxels. The second row (*Figure 4a1-c1*) presents the solid textures obtained with the ML_H1a heuristic. This heuristic performs well, but not strong enough to capture the interactions between pixels (as shown in *Figure 4c1*), a possible solution being to increase the neighborhood size. Using the ML_H1b heuristic (*Figure 4a2-c2*), undesired artifacts are seen on almost all the results. But the averaging operation is not the winning strategy: the voxels are faced with grey-levels that do not exist in the 2D sample, disrupting the image-based synthesis process and producing broken textures. For the last two rows of *Figure 4*, the update voxel value is the value that maximizes a function of the orthogonal views 2D estimations. ML_H2a heuristic yields promising results in terms of structure conservation, but undesired artifacts still come into sight (visible on *4a3* and *4c3*). The LCPDF estimation is not able to capture the pixels correlations under the employed framework, unsuitable for characterizing all the orthogonal views. The ML_H2b heuristics (last row in *Figure 4*) runs better, being able to capture the visual characteristics of the input texture by maximizing the product function of each view's LCPDF. This product based heuristic is stronger (in terms of capturing pixel interactions) than all the others, even at a relative small neighborhood size. Using this heuristic as a decisional measure for the output voxel value, the synthesis provides more than satisfying results. The synthetic solid textures (*Figure 4a4-c4*) are smooth, almost artifact–free (except some small structure

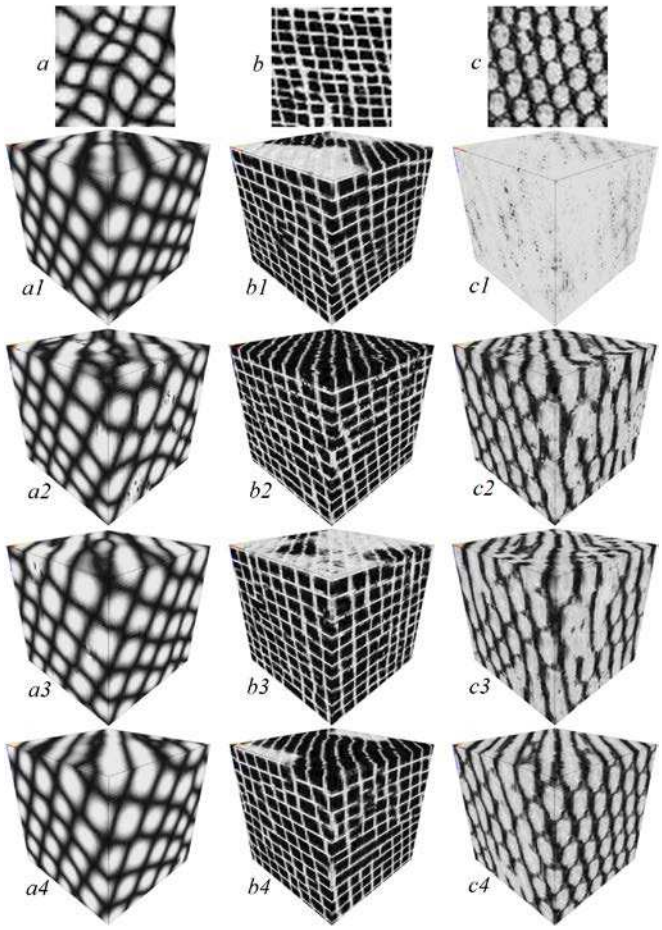

Figure 4 – *ICM based volumetric results: each column shows from top to bottom the results obtained by synthesizing the 1st row texture, using ML_H1a heuristic (the 2nd row), using ML_H1b heuristic (the 3rd row), using ML_H2a heuristic (the 4th row) and ML_H2b heuristic (the 5th row).*

defects), looking very similar to the sample textures giving the impression that they were produced by the same process.

The above remarks justify the choice of using ML_H2b for the next operations as being the most convincing heuristic, representative for the maximum-likelihood based approach.

B. Temperature decreasing schema

The pixel temperature function influences the duration of the ICM relaxation process, controlling the cooling rate. The process starts at high temperatures, and is iteratively relaxed with the decreasing of the temperature. Two aspects drew our attention – the

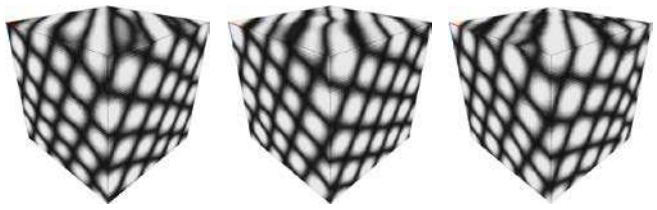

Figure 5 – *Results of different ICM temperature decreasing strategies: from left to right, the first two results obtained by accomplishing 100% of the relaxation process but by decreasing the pixel temperature with $\xi=-1$ and $\xi=-3$, while the 3rd block is obtained with a $\xi=-3$ by stopping the synthesis process after reaching 95% from the initial global temperature.*

speed of the *temperature dropping-off* and the *relaxation duration*. The pace of the cooling schedule is determined by the parameter ξ as described in (8). A small value assures a good convergence but implies long relaxation sequences, while a higher value allows a faster process but not necessarily with satisfying results, incapable of capturing the spatial correlations. To get satisfactory results in a reasonable time, a compromise has to be made regarding the rate of the cooling schedule and the stopping criterion. *Figure 5* shows that reducing the number of iterations by using ξ=-3 and stopping at 95% of the initial global temperature can still produce satisfactory results. The impact of ξ is reflected in a more accelerated decrease of the global temperature. A more important ξ (a smaller value) allows reaching a same global temperature level in fewer iterations. Stopping the synthesis process at 95% from the initial global temperature, the number of iterations is reduced even more (roughly from 18 to 6).

C. Common parameters influence

The synthesis process is sensitive to the *neighborhood system*. The neighborhood size has to be large enough to capture the spatial interactions between neighbors. With the increase of the neighborhood, better are the results but higher will be the computational time. To capture better the textural patterns, multi-resolution is used, the number of scales being questioned. The above parameters influence is discussed in [40], mentioning here only the corroborated effect of the up-sampling strategy with the neighborhood size (treated in *Figure 6*). The 8/8 up-sampling does not assure a required degree of diversity, reproducing the values from the lowest resolution level. If the lowest resolution is inappropriate (i.e. structure misplaced, uniform values) the ICM guided synthesis process propagates errors through all the scales. To assure disparities while passing between scales, a 1/8 up-sampling pays attention to the information from the previous scale and also to a certain randomness induced by initialization.

D. The choice of the relaxation algorithm

The results obtained with the deterministic relaxation and with the stochastic algorithm are roughly similar in terms of texture quality. But the ICM based synthesis implies a more intricate analysis over the pixel confidence function and the scales handling. Additionally, the deterministic relaxation based process is computationally expensive performing LCPDF estimations for every level available in the input image. The input textures should have their dynamics reduced to assure a reasonable computational

Figure 7 – *Volumetric results obtained with different relaxation algorithms over the ML_H2b heuristic: each example contains in the middle the 2D sample texture, on the left the result obtained using the deterministic relaxation algorithm and on the right, the result corresponding to the simulated annealing based synthesis process.*

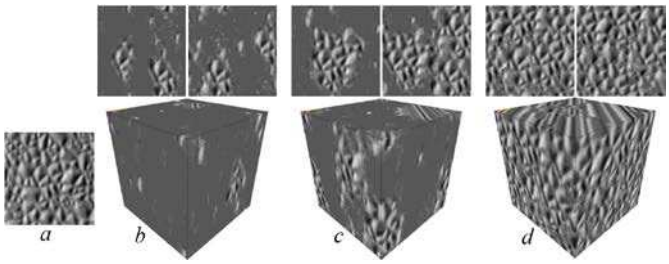

Figure 6 – *The mutual influence of the neighborhood size and the up-sampling strategy on synthesizing the 2D sample in (a): (b) is obtained using a 9×9 neighborhood and an 8/8 up-sampling, (c) is obtained by using a 9×9 neighborhood size but with an 1/8 up-sampling and the result in (d) is obtained with a 11×11 neighborhood corroborated with 1/8 up-sampling strategy; above each block are represented two 2D cuts from the 3D block.*

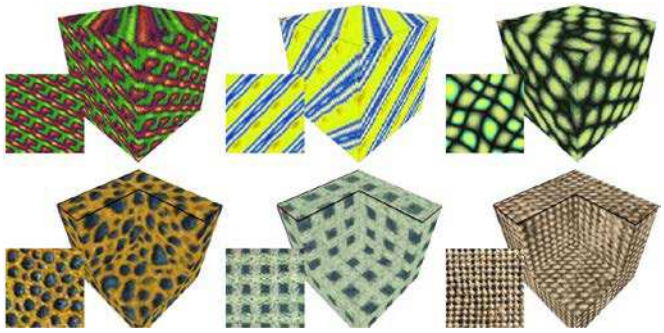

Figure 8 – *Volumetric color results obtained with the stochastic relaxation based algorithm and the ML_H2b heuristic decision.*

time. Moreover, to show some drawbacks of the deterministic algorithm relative to the stochastic relaxation, some synthetic results obtained from sample textures reduced to 32 grey-levels are compared in *Figure 7*. The ICM based synthesis process reduces significantly the texture dynamics maintaining only the most present (i.e. the most probable) grey-levels of the 2D sample, making simpler the 3D texture. On the other hand, the stochastic relaxation allows by its nature a greater diversity preserving better the texture structure during the synthesis process, if a sufficient number of random draws is realized.

E. Color texture synthesis

All the above considerations are valid for the color texture case, the proposed algorithms being able to synthesize color textures. Indeed, the color extension is straightforward since it comes down to enlarging the feature space in which the LCPDF non parametric estimation is performed, taking 3 dimensions instead of 1 to represent each voxel. Unlike [26], the color channels (i.e. red, green, blue) are treated together so that undesired artifacts caused by channels handling are eliminated. The color texture adapted algorithm is applicable both with the deterministic and the stochastic relaxation algorithm. When using the deterministic algorithm, the LCPDF estimations for all the available input color texture values are prohibitive, the computational time being already high for the grey texture case. Indeed, for a 64^2 pixel input color texture, the set of possible colors is almost as big as the number of pixels i.e. 64^2 distinct values, unlike the grey level case when there are less than 256 grey-levels. The solution adopted here is to use the stochastic relaxation assuring that a sufficient high number of random draws is achieved

within a reasonable time complexity. The results in *Figure 8* illustrate the capacity of our simulated annealing based algorithm to synthesize volumetric color textures, also showing the blocks inner structure (2nd row in *Figure 8*).

VI. FIXED-NEIGHBORHOOD SEARCH VS. LIKELIHOOD MAXIMIZATION

This section aims to be a qualitative and a quantitative comparison of the results obtained with the neighborhood-search based synthesis approaches and with the maximum-likelihood based methods. To ease the comparative study only two foremost approaches are used. These are the ML_H2b method based on simulated annealing (motivated by the previous sections) and the fixed-neighborhood search based approach proposed by Kopf *et al.* [17] (as it was proved in [20] and [40] to provide equivalent or better results compared to [16] and [18]) tagged as NP_K approach.

Some synthesis results are shown in *Figure 9*, placing jointly the NP_K and the ML_H2b volumetric textures obtained from both grey-level and color textures.

For the textures shown in *Figure 9a-c* the results based on the ML_H2b heuristic seem to be better than the ones obtained with the NP_K approach, mostly in terms of structure preservation. The texture *9c* shows the NP_K's difficulties in sustaining the structure. A known characteristic of the neighborhood search based algorithms is that they are sub-optimal, leading to synthesis results that suffer from repetitive patterns and from more ordered features than the sample [18,20]. This phenomenon is proved by the NP_K result in *9a,b,* but attenuated when using the maximum likelihood based approach.

The solid textures in *9a-g* obtained with our proposed heuristic and the stochastic relaxation are of satisfying quality, both in structure and dynamics preservation relative to the input samples. For many results, as in *9d-g*, the NP_K and ML_H2b approaches perform quite similarly, hardly differentiating from each other, still being pleasing results.

Figure 9h,i illustrates NP_K results that are better than the one obtained with ML_H2b, even if the dynamics seem to be well conserved. In this case, our proposed heuristic is not strong enough to capture the structure of the sample texture. The neighborhood-search based synthesis approach reproduces in the volume the sample pattern.

Figure 9 – *Volumetric results: each triplet contains in the middle the 2D sample, on the left the 3D texture obtained using NP_K approach and on the right the 3D block obtained by using our proposed ML_H2b heuristic with the stochastic relaxation algorithm.*

Lastly, in *9j-l* the results are bad for both of the approaches, incapable of capturing the sample structure in the synthetic result, while in *9l* both methods seem to provide a synthetic texture more regular than the sample.

The above concurrent process based on our human perception shows that the algorithms provide satisfying results, seeming that the 3D results and the sample textures have been created by the same underlying process. However, beyond the traditional subjective evaluation of the synthesized textures, an objective analysis is required. We propose a quantitative benchmark which consists in comparing the input and output texture characteristics following the quantitative valuation methodology described in [20]. We are focusing on evaluating the results by taking into consideration 1st and 2nd order statistics and the patterns morphology, in particular the local orientations. The graphs in *Figure 10* summarize our objective evaluation on a set of fifteen textures, the quantitative indicators generally confirming our subjective remarks. For instance, the comparison of the results dynamics for the grey-level sample textures (textures *c,f,g,n,o* in *Figure 10*) shows that the NP_K approach alters the dynamics because of the average operation it involves [17], while the ML_H2b method updates the output voxels directly with input pixels. To characterize the structure preservation, the color histogram comparison doesn't provide sufficient information. For the results in *Figure 9* the dynamics seems to be similar, but the textures obviously show different patterns. The comparison based

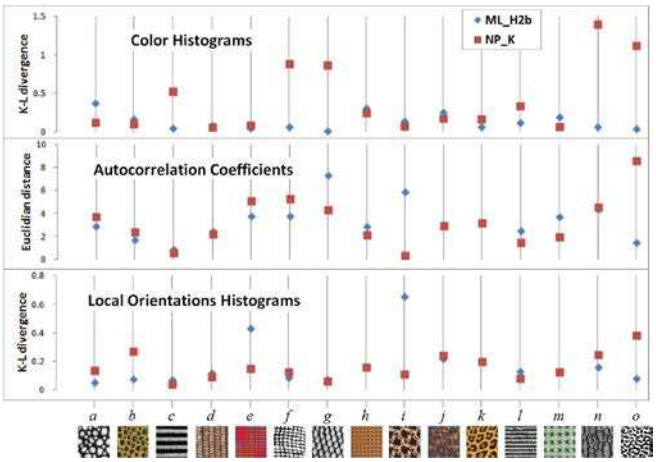

Figure 10 – *Different indicators of the objective comparison of the 3D textures obtained by synthesizing the sample textures (bottom row) with the ML_H2b and the NP_K approach: color histograms, autocorrelation coefficients and local orientations. The procedure consists in computing the Kullback-Leibler divergence between the histograms or the L2 norm between the autocorrelation coefficients of the input and the output texture. Local orientations are computed using the structure tensor (see [20,40] for details). Grey level autocorrelations are considered for 32×32 lags.*

on the local orientations histograms in *Figure 10* detects these differences, proving the NP_K incapacity (for *9a,b*) or the ML_H2b incapacity (for *9i*) to preserve the structure of the sample texture.

From a computational cost point of view, the ML_H2b approach based on simulated annealing, needs long relaxation steps, estimating the LCPDF at each random draw. A high number of draws is required to provide smooth results. If one searches for a fast way to generate textures (not necessarily with the best result), the fixed-neighborhood search based approaches are of interest. For example, using a machine operating at 2.7 GHz to generate a 64^3 voxel volumetric texture from a 64^2 pixel color sample, using a 11×11 neighborhood, 3 scales, the time required by the ML_H2b relaxation has an order of almost 48 hours, while the NP_K based results are obtained in a few minutes. A future development requires the algorithm acceleration, attainable by algorithm parallelization. One other possible way is to accelerate the estimation of the LCPDF by using a small but sufficient proportion of the input data. This can be achieved for example by using a tree structured vector quantization technique [22] or by using a faster kernel density estimator as it was done for the 2D case in [36]. However, it should be noted that such acceleration, by its sub-optimal nature, may lead to losing the benefits of the stochastic relaxation algorithms (especially simulated annealing).

VII. About 2D/3D inference

The solid texture should exhibit a plausible structure in accordance with the reality. Each of its 2D slices should be consistent with the 2D sample. Generating a volume from a 2D image is an under-constrained problem, with multiple solutions, not all of them equally possible. So, the algorithms must be adapted to the input image structure to be able to 'invent' (i.e. infer) detailed 3D structure from a 2D image and to create qualitatively accurate and visually pleasing results. Depending on the texture

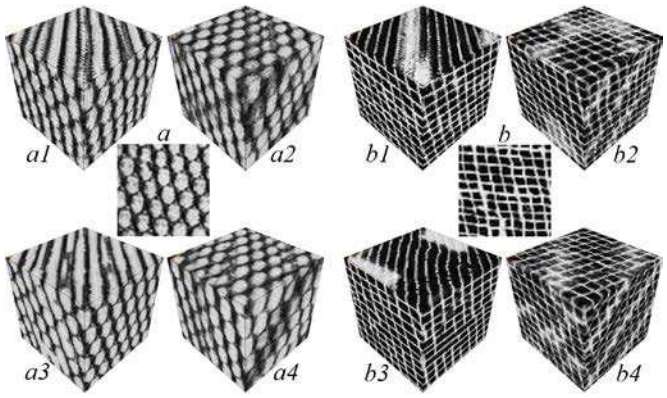

Figure 11 – *3D blocks obtained by synthesizing the 2D samples (in a,b) by constraining 2 views (with NP_K in a1,b1 and with ML_H2b in a3,b3) or by constraining 3 views (NP_K in a2 and b2; ML_H2b in a4 and b4).*

configuration (e.g. asymmetry, homogeneity, granularity etc.), the synthesis has to be constrained by two or three orthogonal views of the 3D block.

Disposing of only a single sample as source of synthesis, the same image is used to constrain the orthogonal views of the block, coercing the front view, side view or top view to match the sample. The number of constrained orthogonal views depends on the sample texture structure. For example, for the anisotropic textures in *Figure 9c,l* showing a lamellar organization, it is not possible to constrain three orthogonal views in the synthesis process by a unique 2D sample, as it does not provide enough accurate information to reproduce a plausible volume. For this type of textures it makes sense to constrain only two views, knowing that their underlying process consisted in layers stacked one above the other. Even when not disposing of the texture for the 3rd view, and using a same texture as constraint for two views, the synthesis process does a good job, indirectly inferring the 2D textural information at the 3D level, both for NP_K and ML_H2b.

The textures in *Figure 11* are susceptible to using two and three views, obtaining in both cases satisfying volumetric results in terms of resemblance between slices and the sample, while preserving a coherent 3D context. This is even better illustrated by the 3D rendering in *Figure 12*. Constraining two orthogonal views, the perceived patterns have an elongated tubular, cylinder shape. Instead, when using the same sample as constraint for three views, the patterns are bubble or cube shaped. Slicing the block in frontal, lateral and from atop, the obtained 2D textures look similar to the sample, the same remark being valid when

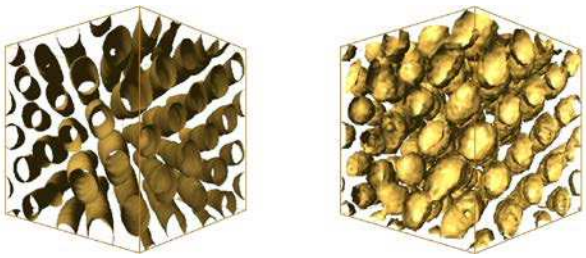

Figure 12 – *Volume rendering of the solid textures from Figure 11a3 (on the left) and 11a4 (on the right).*

slicing frontally and laterally the block obtained by constraining only 2 views.

Depending on the texture consistency and on the user's demand, 3D information can be inferred from 2D observations, constraining two, three or even more views (if appropriate samples are available) leaving the inference discussion open for future considerations.

VIII. CONCLUSION

We have developed a novel probabilistic non-parametric algorithm for the synthesis of volumetric textures using a single 2D sample. This algorithm is based on a MRF conceptualization, describing the interactions between pixel values in terms of the LCPDF i.e. the local conditional probability density function. Based on a 2D paradigm, our original 3D extension relies on giving a new description to the LCPDF of a site in 3D through 2D based heuristics. Notable remarks are made on how to use better the voxel temperature function involved by the synthesis deterministic relaxation process and how to adequately use the stochastic relaxation especially in the color texture case. As well, we propose different strategies for the scales handling policy. All these procedures have been materialized successfully on a set of significantly different textures, identifying the strengths and the weaknesses of the synthesis approaches. The quality of the results is highly encouraging, in terms of structure and dynamics, Moreover, when compared both visually and objectively with a fixed-neighborhood search based algorithm, the likelihood-maximization approach proved to behave significantly better for a number of structured textures. Taking into consideration the increased computational load, the proposed approach may appear as interesting alternative to classical reference algorithms especially when looking to reproduce the texture structure as faithfully as possible.

REFERENCES

[1] L.-Y. Wei, S. Lefebvre, V. Kwatra and G. Turk. State of the Art in Example-based Texture Synthesis. *EUROGRAPHICS 2009 State of the Art Reports (STARs)*.

[2] K. Perlin. An Image Synthesizer. *Computer Graphics (SIGGRAPH Proceedings*), volume 19, pp. 287–296, July 1985.

[3] J.-P. Lewis. Algorithms for Solid Noise Synthesis. *Computer Graphics (SIGGRAPH Proceedings)*, volume 23, pp. 263–270, July 1989.

[4] A. Lagae, S. Lefebvre, G. Drettakis and P. Dutre. Procedural noise using sparse Gabor convolution. *In SIGGRAPH*, 28(3), August 2009.

[5] Y. Chen and H.-S. IP. Texture evolution: 3D texture synthesis from single 2D growable texture pattern. *The Visual Computer*, Vol. 20, No. 10, pp. 650-664, Dec. 2004.

[6] S. Owada, F. Nielsen, M. Okabe, and T. Igarashi. Volumetric illustration: designing 3D models with internal textures. *ACM Transactions on Graphics*. 23, 3, pp.322–328, 2004.

[7] N. Pietroni, M. Otaduy, B. Bickel, F. Ganovelli and M. Gross. Texturing internal surfaces from a few cross sections. *Computer Graphics Forum* 26, 3, pp.637–644, 2007.

[8] K. Takayama, M. Okabe, T. Ijiri and T. Igarashi. Lapped solid textures: filling a model with anisotropic textures. *Proceedings of ACM SIGGRAPH*, Vol. 27, 3, 2008.

[9] E. Praun, A. Finkelstein and H. Hoppe. Lapped textures. *The 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., pp. 465–470, 2000.

[10] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. *In SIGGRAPH* , pp. 229–238, New York, NY, USA, 1995.

[11] J. Portilla and P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision* 40, 1, pp. 49–70, 2000.

[12] J.-P. Da Costa and C. Germain. Synthesis of solid textures based on a 2D example: application to the synthesis of 3D carbon structures observed by transmission electronic microscopy. *Proc. of SPIE*, vol. 7538, *Image Processing: Machine Vision Applications III*, pp.10, 2010.

[13] D. Ghazanfarpour and J.-M. Dischler. Spectral analysis for automatic 3D texture generation. *Computers and Graphics* 19, 3, pp. 413–422, 1995.

[14] D. Ghazanfarpour and J.-M. Dischler. Generation of 3D texture using multiple 2D models analysis. *Computer Graphics Forum 15*, 3 (Proc. Eurographics), pp. 311–323, 1996.

[15] J.-M. Dischler, D. Ghazanfarpour and R. Freydier. Anisotropic solid texture synthesis using orthogonal 2D views. *Computer Graphics Forum 17*, 3 (Proc. Eurographics), pp. 87–96, 1998.

[16] L.-Y. Wei and M. Levoy. Texture synthesis from multiple sources. *ACM SIGGRAPH Sketches & Applications*, New York, 2003.

[17] J. Kopf, C.W. Fu, D. Cohen-Or, O. Deussenn, D. Lischinski and T.T. Wong. Solid Texture Synthesis from 2D Exemplars. *ACM SIGGRAPH*, TOG, vol. 26, issue 3, 2007.

[18] J. Chen and B. Wang. High quality solid texture synthesis using position and index histogram matching. *The Visual Computer*, vol.26, pp. 253-262, 2010.

[19] V. Kwatra, I. Essa, A. Bobick and N. Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics* 24, 3 (Proc. SIGGRAPH), pp.795–802, 2005.

[20] R. Urs, J. -P. Da Costa, J. -M. Leyssale, G. Vignoles and C. Germain. Non-parametric synthesis of laminar volumetric textures from a 2D sample. *In Proceedings of British Machine Vision Conference*, pp.54.1-54.11, 2012.

[21] J. C. Prieto, C. Revol-Muller, F. Peyrin, P. Camelitti and C. Odet, 3D Texture synthesis for modeling realistic organic tissues. *VISAPP 2012. International Conference on Computer Vision Theory and Applications*, 24-26, pp.60-65, Rome, Italy, February 2012.

[22] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. *In SIGGRAPH, 27th International Conference on Computer Graphics and Interactive Techniques*, pp. 479-488, 2000.

[23] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo and H. Shum. Synthesis of bi-directional texture functions on arbitrary surfaces. *In SIGGRAPH: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. ACM Press, pp.665–672, 2002.

[24] J. Han, K. Zhou, L. -Y. Wei, M. Gong, H. Bao, X. Zhang and B. Guo. Fast example - based surface texture synthesis via discrete optimization. *Visual Computer* 22(9), pp.918–925, 2006.

[25] S. Lefebvre and H. Hoppe. Appearance - space texture synthesis. *In ACM Transactions on Graphics*, 25(3), pp.541-548, 2006.

[26] X. Qin and Y. -H. Yang. Aura 3D textures. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, 2, pp. 379-389, 2007.

[27] J. Dischler and D. Ghazanfarpour. Interactive Image - Based Modeling of Macrostructured Textures. *Computers & Graphics*, pp. 66–74, 1999.

[28] R. Jagnow, J. Dorsey and H. Rushmeier. Stereological Techniques for Solid Textures. *In: SIGGRAPH '2004*, pp.329–335, 2004.

[29] J.W. Chiou and C.K. Yang. A New Algorithm for Solid Texture Synthesis. *Lecture Notes in Computer Science 4292/2006 (Advances in Visual Computing)*, pp. 780-789, 2006.

[30] R. Paget and I. Longstaff. Texture synthesis via a noncausal nonparametric multiscale Markov random field. *In IEEE Transactions on Image Processing* 7, 6, pp. 925–931, 1998.

[31] J. E. Besag. On the statistical analysis of dirty pictures. *Journal of The Royal Statistical Society*, vol. B-48, pp. 259-302, 1986.

[32] J. E. Besag. Spatial interaction and the statistical analysis of lattice systems. *In Journal of the Royal Statistical Society*, series B, vol. 36, pp. 192-326, 1974.

[33] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721-741, 1984.

[34] P. Clifford. Markov Random Fields in Statistics. *In Disorder in Physical Systems: A Volume in Honour of John M. Hammersley*, eds. G. Grimmett and D. Welsh, Oxford University Press, pp.19–32, UK, 1990.

[35] D. Geman. Random fields and inverse problems in imaging. *In Lecture Notes in Mathematics*, vol. 1427, pp. 113-193. Springer-Verlag, 1991.

[36] A. Sinha and S. Gupta. A Fast Nonparametric Noncausal MRF-Based Texture Synthesis Scheme Using a Novel FKDE Algorithm. *IEEE Transactions on Image Processing*, vol. 19, no. 3, 2010.

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

[38] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, vol. 21, pp. 1087-1091, 1953.

[39] C. Bouman and B Liu. Multiple resolution segmentation of textured images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13 -2, pp. 99-113, 1991.

[40] R. Urs. Non-parametric synthesis of volumetric textures from a 2D sample. *PhD thesis no 4773*, *University of Bordeaux,* 2013.