

# EXEMPLAR-BASED TEXTURE SYNTHESIS USING TWO RANDOM COEFFICIENTS AUTOREGRESSIVE MODELS

AYOUB ABDERRAZAK MAAROUF<sup>✉,1</sup>, FELLA HACHOUF<sup>1</sup> AND SOUMIA KHARFOUCHI<sup>2</sup>

<sup>1</sup>Laboratoire d'Automatique et de Robotique, Département d'Electronique Université des frères Mentouri, Constantine 1 Algérie, <sup>2</sup>Département de médecine, Bon Pasteur Chalet des Pins, Université Constantine 3 Algérie

e-mail: ayoub.maarouf@umc.edu.dz, hachouf.fella@gmail.com, S.kharfouchi@yahoo.fr

(Received January 25, 2023; revised March 23, 2023; accepted March 23, 2023)

## ABSTRACT

Example-based texture synthesis is a fundamental topic of many image analysis and computer vision applications. Consequently, its representation is one of the most critical and challenging topics in computer vision and pattern recognition, attracting much academic interest throughout the years. In this paper, a new statistical method to synthesize textures is proposed. It consists in using two indexed random coefficients autoregressive (2D-RCA) models to deal with this problem. These models have a good ability to well detect neighborhood information. Simulations have demonstrated that the 2D-RCA models are very suitable to represent textures. So, in this work, to generate textures from an example, each original image is splitted into blocks which are modeled by the 2D-RCA. The proposed algorithm produces approximations of the obtained blocks images from the original image using the generalized method of moments (GMM). Different sizes of windows have been used. This study offers some important insights into the newly generated image. Satisfying obtained results have been compared to those given by well-established methods. The proposed algorithm outperforms the state-of-the-art approaches.

Keywords: exemplar based method, GMM, local approximated images, texture synthesis, 2D-RCA models.

## INTRODUCTION

Over the last four decades, many texture descriptions have been proposed in the field of digital image processing (Haralick, 1979; Akl, 2016). From a structural standpoint, texture derives from spatial variations in the grey levels of pixels. It is related to a wide range of natural phenomena that exhibits repeating patterns with some randomness. Textures are often classified according to their degree of regularity, which ranges from regular to stochastic. However, one of the most important features used to distinguish various regions of an image is texture. Versatility of textures applications has been shown clearly: Pattern recognition, image classification, image segmentation (Salmi *et al.*, 2021), image enhancement, image compression, fault detection (Sun *et al.*, 2009), medical imaging diagnosis (Duncan and Ayache, 2000), and analysis of material structure are examples of generic (Da Costa *et al.*, 2015) activities that can be focused on the concept of texture.

Texture synthesis aims to infer a generating process from an example texture, allowing the production of an infinite number of new samples of that texture. Human analysis is typically used to assess the consistency of the synthesized texture. And, textures are considered adequate if a human observer cannot distinguish between the original and the synthesized

textures. However, this research is challenging due to the lack of a standard description for the concept of texture. The ability of parametric approaches to analyze and generate structures is constrained, but they do well with stationary and uniform textures. By using patches that are large enough to capture the local structures, nonparametric approaches can handle small scale structures. In this paper, we are interested in approaches based on a statistical or stochastic characterization of suitable local neighborhood properties for natural textures. The statistical methods study investigates the relationships between a pixel and its neighbors. They are particularly adapted to the stochastic textures analysis. The statistical approaches consist in correlation functions, frequency domain analysis, edges operators, gray-level co-occurrence matrix (GLCM) (Haralick *et al.*, 1973), Gabor textural features, autoregressive models and Markov processes (Reis and Taşdemir, 2011).

In textural modeling, the texture is assumed to realize a stochastic process. Therefore, parameters of these models are considered to be the texture characteristics. In autoregressive modeling, it is assumed that there is an interaction between the grey level of each pixel in the image and its neighbors.

Exemplar-based texture synthesis methods have led to the development of numerous techniques for creating textures that are visually identical to input texture

samples (Wei *et al.*, 2009). The ability of parametric approaches to analyze and generate structures is constrained, but they do well with stationary and uniform textures. By using patches that are large enough to capture local structures, nonparametric approaches can handle small scale structures.

Recently several convolutional neural networks (CNN)-based synthesis techniques have been proposed and presented to produce state-of-the-art results (Wang *et al.*, 2021; Ulyanov *et al.*, 2016). However, deep learning has not achieved the same level of success in texture representation as it has performed in other tasks (Basu *et al.*, 2018). Thus, it is a drawback of deep learning given that texture is an inherent property of objects. And it is a critical descriptor for many computer vision applications.

Therefore, in the present study, we are interested in a new method that has succeeded in great representing a wide range of textures, namely the 2D-RCA models. Indeed, authors in (Bouleznadjel *et al.*, 2015), have generated from several first-order 2D-RCA models some synthetic textures and proved that they are quite similar to appropriate natural textures.

Motivations of considering some textures as a realization of a stationary 2D-RCA process of the first order, three arguments justify this assertion: Simulation tests have demonstrated that this model is able to represent a wide range of textures. It should be noticed that the 2D-RCA models contain a large number of benefits.

It is characterized by areas clustering with strong gray-scale variations, followed by areas with moderate or weak changes.

The first order has been chosen because first-order models are commonly used in practice. That is, to represent the value of a specific site, only the pixel's immediate neighbors are considered.

The framework of the proposed method is shown in Fig.1.

The paper is arranged as follows: Section 2 presents related works. Section 3 introduces briefly the 2D-RCA models theory as well as the generalized method of moments which has been used to estimate the 2D-RCA parameters to produce the synthesis texture. The proposed approach is described in section 4. Section 5 shows experimental results. Finally, conclusion and future work are given in the last section.

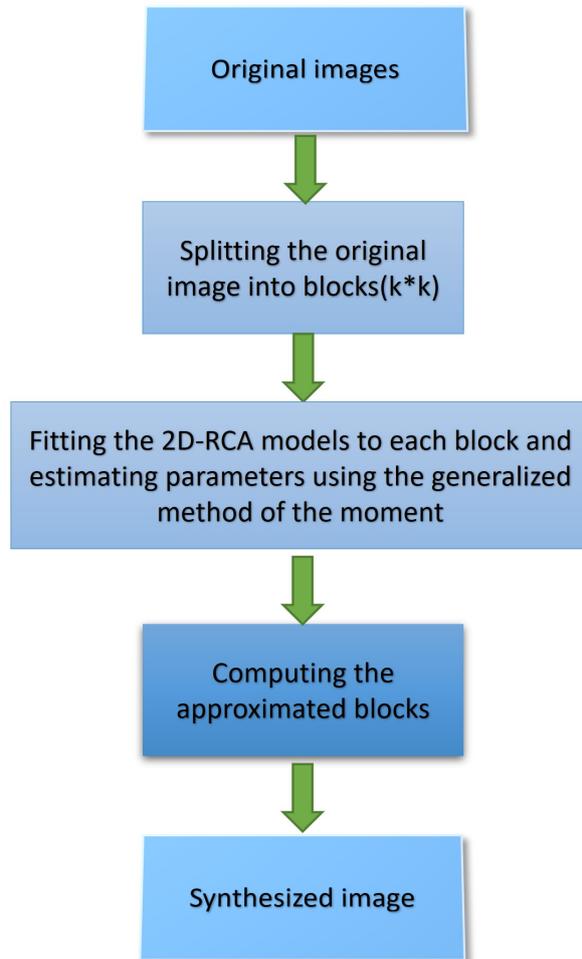


Fig. 1: Block diagram of the proposed method

## RELATED WORKS

Over the last few decades, research on texture synthesis has resulted in several techniques divided into procedural, exemplar-based, model-based texture synthesis methods, and deep learning models.

**1. Parametric synthesis:** Procedural texture synthesis aims to create textures using mathematical functions or algorithms with a fixed computational cost. As a result, procedural methods are well-suited tools for generating objects texture in simulated worlds, such as video games (Musgrave *et al.*, 1994). These methods work by transforming a collection of pre-defined signals into the desired texture. They are typically used to create highly structured textures (like pavement) or unstructured textures (like Worley noise). Other noise functions, such as the Gabor noise, have been suggested. (Lagae *et al.*, 2009; Galerne *et al.*, 2010).

**2. Model-based texture synthesis:** These techniques aim to create probabilistic models that

can be used to both define and synthesize texture. The model's parameters should capture the texture's essential visual characteristics. In the literature, there are various models such as (Chellappa and Kashyap, 1985; Clark *et al.*, 1987; Cross and Jain, 1983). It should be noticed that the models can be learned from an example image, which means that some of these approaches are quite similar to exemplar-based synthesis algorithms. Some synthesis techniques can fall into different classification categories. Synthesis, for example, can be done from models using a procedural method.

**3. Exemplar-based texture synthesis:** One or more example textures are needed as inputs for these methods. The majority of them create textures by directly copying pixels or patches (sub-images) from the input images. As a result, these methods include creating a new texture as close to the texture models (input) or exemplar as possible. They consist of three leading families that can create various textures. (Qian *et al.*, 2018; Portilla and Simoncelli, 2000):

### 3.1. Parametric texture synthesis by analysis:

This group of methods is focused on a statistical description of the input texture. First, a new texture is created to impose a collection of statistical constraints (textural signatures or parameters) on the output texture based on the input image. To obtain the used parameters in the synthesis process, characteristics of the input sample are extracted. After that, the synthetic texture is compared to the exemplar one to ensure that the two textures are visually identical. These methods create the texture pixel by pixel while preserving the local texture's coherence with its surroundings (Tong *et al.*, 2002; Hertzmann *et al.*, 2001; Zelinka and Garland, 2002). Most of them use Markov field theory, which models the consistency of realizing a local and stationary phase. Most of these approaches combine practical search algorithms with a multi-scale implementation capable of expressing the models at different scales without dramatically raising the computational load to ensure a fair computational cost.

**3.2 Patch-based texture synthesis:** Patch-based synthesis has emerged as a computational complexity improvement over pixel-based approaches, with a high computational cost, particularly when reproducing highly structured images. The general process entails choosing the most similar patch in the exemplar to the current neighborhood in the output texture. Optimization techniques are used to reduce edge defect artifacts. (Efros and Freeman, 2001). It is interesting to note that the exemplar-based family encompasses most synthesis techniques.

**4. Texture synthesis using Deep learning approaches:** Recently several convolutional neural

networks (CNN)-based synthesis methods have been proposed (Wang *et al.*, 2021; Xie *et al.*, 2018).

Gatys *et al.* (Gatys *et al.*, 2015b) introduced another kind of texture synthesis technique that depends on a convolutional neural network. This approach consists in producing new parameters. Authors in (Gatys *et al.*, 2015b) have combined features space of a convolutional neural network with the principle of spatial summary statistics on features responses. In this way, acquired texture models parameterized spatially invariant portrayals based on the various leveled architecture of the convolutional neural network. However, one drawback of these methods is the challenge of effectiveness for large-scale regularity, as shown in Fig.2.

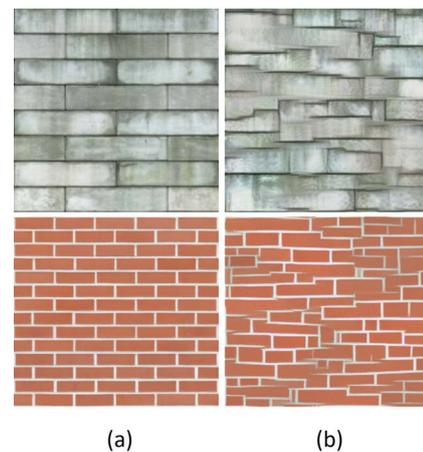


Fig. 2: *Texture synthesis using CNNs according to (Gatys et al., 2015b). a: an exemplar texture. b: synthesized result using the CNN model*

In addition, another study of texture synthesizing relies on the use of a generative adversarial neural network (Xian *et al.*, 2018). Authors proposed a deep generative network that can synthesize multiple textures' outputs in a single network. Inspired by the concept of up-convolutions, the network's architecture generates texture images from a noise vector and a selection unit as inputs. The selection unit is a one-hot vector with each bit representing a texture form that gives users a control signal to move between different textures to synthesize. Recently, in (Wang *et al.*, 2021) a novel texture model, called conditional generative ConvNet (cgCNN) has been proposed by combining deep texture statistics with the probabilistic framework of generative ConvNet (gCNN) (Xie *et al.*, 2016). Given a texture, cgCNN uses deep statistics of a trainable ConvNet to build an energy-based conditional distribution, which is subsequently trained via the maximum likelihood estimation (MLE). By sampling from the learned conditional distribution, new textures can be created. cgCNN learns the

weights of the ConvNet for each input exemplar, unlike prior texture models that relied on pre-trained ConvNets. As a result, it has two major advantages: 1) It enables the creation of unified images. 2) It is easy for the sampling method to escape from local minimums. Thus it can synthesize textures with non-local structures without using additional penalty terms.

## PROPOSED METHOD

Example-based texture synthesis consists in the most used and researched texture creation algorithms. These approaches take a real-world image as an input. And, they attempt to produce a new texture from it. The objective of these methods is to generate a new texture to be perceptually similar to the original one. For more than two decades, example-based texture synthesis stands as an active research topic. Up to now, generating an Example-based texture is a challenge. In this work, we present a new texture synthesis technique based on the 2D-RCA models.

It is important to underline that the 2D-RCA parameters have been used in (Boulemdjadjel *et al.*, 2015) as discriminating characteristics in image classification and in a small simulation proving their ability to represent a texture. No real texture modeling has been done to generate real texture images. In this paper, we deal with this problem. Before presenting the proposed method, we present an overview of the construction of the 2D-RCA models.

### Glossary

- Variables in bold are vectors in  $Z^2$ .
- Underlined variables are vectors.
- For each vector  $x(t)$ ,  $\hat{X}(N)$  is the empirical moment.
- For each vector :  
 $\mathbf{s} = (s_1, s_2)$  and  $\mathbf{t} = (t_1, t_2)$ , we write  $\mathbf{s} \ll \mathbf{t}$   
if and only if  $[(s_1 < t_1) \vee (s_1 = t_1) \wedge (s_2 \leq t_2)]$ .
- for  $\mathbf{a}, \mathbf{b} \in Z^2$  such that  $\mathbf{a}=(a_1, a_2)$ ;  $\mathbf{b}=(b_1, b_2)$ ; and  $\mathbf{a} \ll \mathbf{b}$ ,
- $S[a, b]$  is an indexed set defined by:  
 $S[\mathbf{a}, \mathbf{b}] = \{(l, m) \in Z^2 / a_1 \leq l \leq b_1, a_2 \leq m \leq b_2\}$   
by arranging its terms by the lexicography order  $\ll$ .

### 1. Spatial RCA models and parameters estimation

S.Kharfouchi (KHarfouchi, 2012) has introduced the spatial non-linear model which generalizes the

standard random coefficients autoregressive models (RCA) to two dimensions; 2D-RCA. It is generated by:

$$X(t) = \sum_{s \in ]0; P]} a_s(\mathbf{t})x(\mathbf{t}-s) + e(\mathbf{t}), \mathbf{t} \in Z^2 \quad (1)$$

Where  $a_s(\mathbf{t}) = \alpha_s + \beta_s(\mathbf{t})$

For these models, we need the following assumptions:

- $e(\mathbf{t})$ ;  $\mathbf{t} \in Z^2$  is an independent second-order stationary sequence of random variables with mean zero and variance  $\sigma^2$ .
- The  $\alpha_s$ ;  $s \in S]0; P]$  are real constants.
- If  $\underline{\beta}(\mathbf{t}) = (\beta_s(\mathbf{t}); s \in S]0; P])'$  then  $(\underline{\beta}(\mathbf{t}), \mathbf{t} \in Z^2)$  is an independent sequence of  $d \times 1$  random vectors with mean zero and  $E(\underline{\beta} \underline{\beta}') = C$ .
- $\underline{\beta}(\mathbf{t})$  and  $e(\mathbf{t})$  are independent.

The proposed 2D-RCA models in (Boulemdjadjel *et al.*, 2015) are defined on a regular network. They are unilateral by construction; only most of the images we process are acquired with irregular pixels. Fortunately, with the increasing use of computer technology, in at least some situations, data with irregularly spaced pixels may be replaced by a regular grid using image interpolation techniques and resampling programs. As in practice, most spatial models are of the first order; we will focus on the first-order 2D-RCA models given by:

$$X(i, j) = \alpha X(i, j-1) + \beta X(i-1, j) + \gamma X(i-1, j-1) + a_1(i, j)X(i, j-1) + a_2(i, j)X(i-1, j) + a_3(i, j)X(i-1, j-1) + \varepsilon(i, j) \quad (2)$$

Where  $\varepsilon(i, j)$ ,  $(i, j) \in Z^2$  is an independent and identical distributed (i.i.d) sequence of random variables with zero mean and a variance  $\sigma^2$ .

$\alpha$ ;  $\beta$  and  $\gamma$  are constants.  $\{a_l(i, j); (i, j) \in Z^2\}$  are independent sequences of random variables centered with  $E[a_l^2(i, j)] = \eta_l^2$  and  $a_l(i, j)$  are independent of  $\{\varepsilon(i, j); (i, j) \in Z^2\}$  for all  $l = 1, 2, 3$ .

The second order stationarity condition is given by:

$$\begin{cases} \Delta > 0 \\ \frac{1}{2} (\alpha^2 + \beta^2 + \eta_1^2 + \eta_2^2 + \sqrt{\Delta}) < 1 \end{cases}$$

Where :

$$\Delta = (\alpha^2 + \beta^2 + \eta_1^2 + \eta_2^2)^2 + 4(\gamma^2 + \eta_3^2 + 2\alpha\beta\gamma) \geq 0 \quad (3)$$

Under stationary conditions, the 2D-RCA model estimation given by Equation (4) is achieved by the

generalized method of moments (GMM) (Kelejian and Prucha, 1999). Based on the observations  $(X(i,j), 1 \leq i \leq n, 1 \leq j \leq m)$ , the GMM estimator of  $\underline{\theta}=(\alpha, \beta, \gamma)$  is given by:

$$\hat{\underline{\theta}}_{N \times M} = \left( \hat{\alpha}_N, \hat{\beta}_N, \hat{\gamma}_N \right) = \left( \sum_{i=1}^N \sum_{j=1}^M [x(i,j)x'(i,j)] \right)^{-1} \times \sum_{i=1}^N \sum_{j=1}^M x(i,j)X(i,j) \quad (4)$$

where:

$$\underline{x} = (X(i, j - 1); X(i - 1, j); X(i - 1, j - 1)). \quad (5)$$

The (2D-RCA) features are represented by the  $n \times m$  matrix  $\hat{\underline{\theta}}_{N \times M}$  ( $N \times M$  is the used images size).

**2. Texture synthesis algorithm using 2D-RCA modeling:** The 2D-RCA models have two key characteristics. For starters, simulation experiments have shown that these models can represent a wide range of textures. Second, to describe many texture images, the 2D-RCA models do not require a large number of parameters.

In this section, a new algorithm to synthesize texture is presented. This algorithm produces an approximation of the images by using the 2D-RCA models. The algorithm is based on the fact that it can represent any texture images using unilateral first-order 2D-RCA processes.

The original image is divided into squared sub-images of size  $k \times k$ . The first-order 2D-RCA models are fitted to each block. Then, for each local fitted model, an approximated sub-image is formed by using the general method of moments. For all sub-images from each local fitted model, intensities on the boundaries are presented by smoothing the edge between blocks.

Let:

$$X = X(m, n) \quad 0 \leq m \leq M - 1, 0 \leq n \leq N - 1 \quad (6)$$

be the original image.

Consider the approximated image  $\hat{X}$  of  $X$ , of the form:

$$\hat{X}(i, j) = \hat{\alpha}X(i - 1, j) + \hat{\beta}X(i, j - 1) + \hat{\gamma}(1, 1)X(i - 1, j - 1) \quad (7)$$

where  $\hat{\alpha}$ ,  $\hat{\beta}$  and  $\hat{\gamma}$  are the GMM estimates of  $\alpha$ ,  $\beta$  and  $\gamma$  method.

Let

$$4 \leq k \leq \min(M, N)$$

Block size lower then  $(k \times k = 4 \times 4)$  gives non

significant results.

For the sake of simplicity, we assume that images to be processed are ordered so that the number of columns minus one and the number of rows minus one are both multiples of  $k - 1$ ; that is,

$$X = X(m', n') \quad 0 \leq m' \leq M' - 1, 0 \leq n' \leq N' - 1 \quad (8)$$

where:

$$M' = \left[ \frac{M-1}{k-1} \right] (k-1) + 1, N' = \left[ \frac{N-1}{k-1} \right] (k-1) + 1.$$

For all  $i_b = 1, \dots, \left[ \frac{M-1}{k-1} \right]$ , and for all  $j_b = 1, \dots, \left[ \frac{N-1}{k-1} \right]$ , we define the  $(k-1) \times (k-1)$  block  $(i_b, j_b)$  of the

image  $X$  by:

$$B_X(i_b, j_b) = Z(r, s) \quad (9)$$

$$(k-1)(i_b-1) + 1 \leq r \leq (k-1)i_b$$

$$(k-1)(j_b-1) + 1 \leq s \leq (k-1)j_b$$

The method is resumed by the following algorithm and Fig.3

---

Algorithm 1: Texture synthesis using 2D-RCA models

---

**Input:** Original image  $X$ , windows size  $(k \times k)$

**Output:** the approximated image  $\hat{X}$

**Step 1:** Divide the original image into blocks  $B_X(i_b, j_b)$ .

**Step 2:** Model each image block  $B_X(i_b, j_b)$  with the 2D-RCA models by the Eq.2 and the GMM estimators  $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$  given by the Eq.5

**Step 3:** Let  $\hat{Z}$  be the approximated block  $B_X(i_b, j_b)$  by:

$$\hat{Z}(r, s) = \hat{\alpha}Z(r-1, s) + \hat{\beta}Z(r, s-1) + \hat{\gamma}(1, 1)Z(r-1, s-1) \quad (10)$$

**Step 4:** Concatenate all the approximated sub-blocks images  $\hat{Z}(r, s)$ .

**Step 5:** Compute the approximated image  $\hat{X}$  of  $X$  as:

$$\hat{X}_{m,n} = \hat{Z}_{m,n} + \bar{X}$$

## RESULTS AND DISCUSSION

In this section, extensive experimental results are presented to assess the efficiency of the proposed algorithm. The proposed method has been applied to a large number of textures of three different databases including, Brodatz database (Tuceryan and Jain, 1993), DTD (Cimpoi *et al.*, 2014) and the CG-Textures (Sendik and Cohen-Or, 2017).

**DTD dataset :** contains 5640 images. For each one 47 texture classes are available for the 120 image

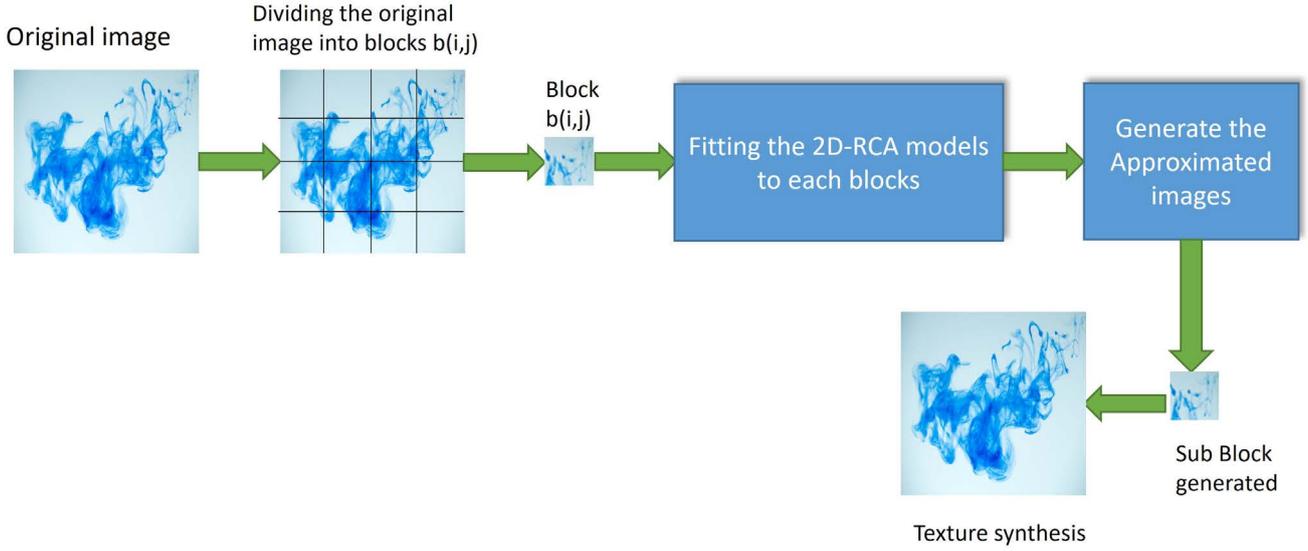


Fig. 3: Proposed method

Table 1: Image quality comparison using different windows size

Window size	8×8			16×16			32×32			64×64		
	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR
image1	0.993	4.543	21.71	0.995	9.421	24.121	0.998	4.639	26.712	0.995	15.765	26.712
image2	0.998	5.456	20.581	0.998	5.885	24.121	0.998	4.064	24.121	0.997	2.458	23.89
image3	0.996	1.423	22.141	0.997	1.4	22.818	0.997	0.825	23.79	0.997	4.45	20.08
image4	0.998	0.93	23.26	0.997	1.425	22.11	0.993	0.745	23.01	0.995	3.17	20.08
image5	0.996	2.5	22.81	0.991	1.5	22.18	0.992	0.489	23.224	0.997	4.067	20.78
image6	0.997	1.543	22.72	0.996	1.4320	23.11	0.999	0.065	23.82	0.992	1.459	21.34

categories. Image sizes range between  $300 \times 300$  and  $640 \times 640$ . It is considered the most challenging data set because it contains large images.

**Broadatz textures:** are the most used texture data set, particularly in the fields of computer vision and signal processing. Because they have been used so frequently in previous texture analysis/synthesis papers, including at least some of them in a texture synthesis study is nearly unavoidable. It consists of 112 textures in grayscale images of various texture types.

**CG-texture databases:** contain color images texture growing from a collection of natural textural pictures, labeled with a variety of human-centered characteristics that are motivated by the perceptual characteristics of textures. Computer vision community has access to this information for research needs.

The proposed method can synthesize textures with high quality and stability. To evaluate the quality of the reconstructed images by the proposed algorithm, three image quality assessment metrics have been used. Structural Similarity Index (SSIM) (Wang *et al.*, 2004), Peak signal-to-noise ratio (PSNR) (Zhang *et al.*, 2011) and Mean square error (Wang and Bovik, 2002)

have been used to quantify the similarity between the original image and the reconstituted one. Best-quality images will have a lower MSE, a high PSNR, and a high SSIM. SSIM, PSNR and MSE are given by:

$$\text{MSE} = \frac{1}{MN} \sum_{n=0}^M \sum_{m=1}^N [\hat{g}(n,m) - g(n,m)]^2 \quad (11)$$

$$\text{PSNR} = 10 \log_{10} (\text{peakval}^2) / \text{MSE} \quad (12)$$

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (13)$$

where  $\mu_x$  and  $\mu_y$  are the local means,  $\sigma_x$  and  $\sigma_y$  are the standard deviations and  $\sigma_{xy}$  is the cross-covariance for images  $x$  and  $y$  sequentially.

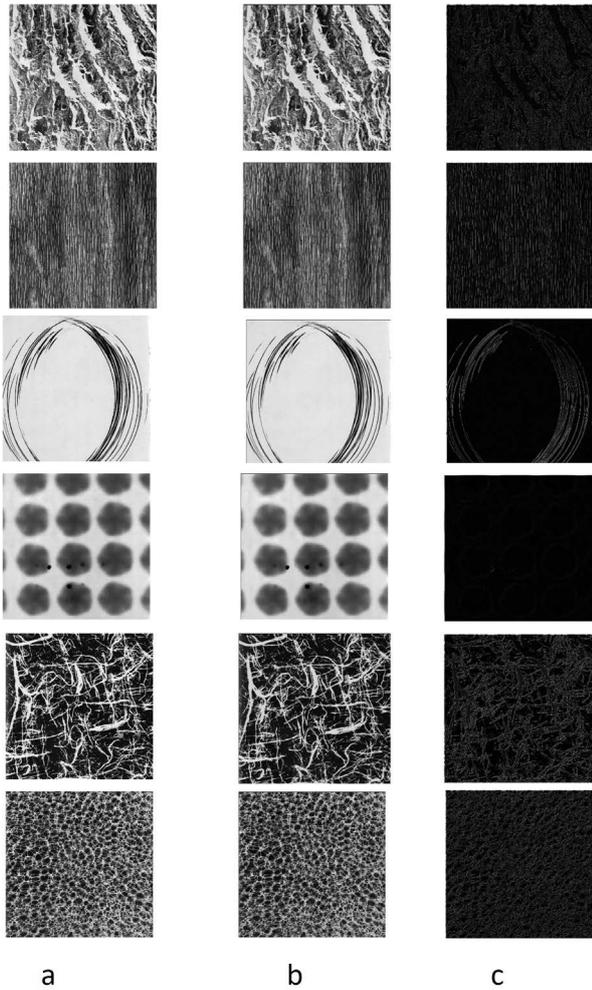


Fig. 4: Original Image (a), (b) generated image by 2D-RCA models, (c) difference between the two images

The first step of the proposed algorithm consists in finding the best window size to give the best results. For this purpose, multiple tests using several windows sizes;  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ , and  $64 \times 64$  pixels, respectively have been conducted. Moreover, the effect on the texture synthesis results is demonstrated. Thus, different image textures from Brodatz database have been tested. It is interesting to remark that using windows size lower than  $5 \times 5$  is inappropriate to characterize a texture. Also fitting the 2D-RCA models to the entire image consists of three parameters that are not enough to model textures images.

Fig.4a represents the original images. Reconstructed images by the proposed method are given by Fig.4b using the window size  $32 \times 32$ . Fig.4c shows the difference between the original and the reconstituted images. For different images, it is observed that the reconstructed images are visually acceptable. Texture structures, boundaries and regions are preserved as it is remarked in Fig.4c.

Table 2: SSIM, MSE and PSNR for Synthesized results of DTD and brodatz databses.

image	SSIM	PSNR	MSE
image 1	0.9995	27.001	0.0025
image 2	0.9997	26.001	0.0023
image 3	0.9998	27.125	0.0084
image 4	0.9994	23.03	0.0897
image 5	0.9993	19.4248	0.0119
image 6	0.9996	22.0443	0.0087
image 7	0.9990	17.25	0.0793
image 8	0.9991	20	0.1528
image 9	0.9990	15.56	0.0278
image 10	0.9996	21.2821	0.0074
image 11	0.9997	22.5054	0.0056
image12	0.9999	26.6825	0.0021
image 13	0.9998	24.8141	0.0033
image 14	0.9996	21.2436	0.0075
image 15	0.9998	22.89471	0.0717
image 16	0.9998	26.7121	0.0021
image 17	0.9996	21.6248	0.0069
image 18	0.9998	28.8630	0.0670
image 19	0.9995	20.0573	0.0993
image 20	0.9998	20.2697	0.0094

Table 1 represents the obtained values of SSIM, MSE, and PSNR for different sizes of windows,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ , and  $64 \times 64$ . According to the given results in table 1 and Fig.4, it is noticed that the best results are shown for a window size of  $32 \times 32$ . Bad results are inherent to the above mentioned remark concerning the window size and the number of parameters representing a texture. So:

- Some regions appear to be homogeneous on a broad scale, implying that they are a realization of a stationary random process, yet heterogeneous on a small scale, implying that the region is non-stationary.
- For small windows, gray levels in different regions are often comparable.

Thus, for the following tests, we adopt a window size of  $32 \times 32$ . Various textures from different image databases have been synthesized to give more value to the proposed method. Fig.5 presents the synthesized results of DTD and brodatz databases. Each panel on the left is the original texture; on the right: the synthesized result using the proposed method. In this experiment, only the results of 20 textures are presented.



Fig. 5: Synthesized results of DTD and brodatz databases by the proposed method. In each panel, Left: original texture, Right: synthesized result.

Table 3: Quality metrics comparison given by (Fig.6)

Window size	Proposed method			Portilla and Simoncelli's			Kaspar et al's			Gatys et al's			the deep correlation synthesized		
	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR	SSim	MSE	PSNR
image1	<b>0.9997</b>	<b>0.0047</b>	<b>23.2990</b>	0.1652	9.2866e+03	9.2866e+03	0.1461	8.5796e+03	8.7961	0.1453	9.3568e+03	8.215	8.4195	0.1483	14.7218
image2	0.9996	0.0068	21.6777	0.1369	1.9630e+03	15.2016	0.1278	1.9195e+03	15.2988	0.0789	1.9321e+03	15.2706	0.0970	2.2149e+03	14.6773
image3	0.9994	0.0120	19.2006	0.1786	1.2643e+03	17.1122	0.1989	1.5741e+03	16.1605	0.0895	15.3198	15.21	0.1256	2.1224e+03	14.8626
image4	0.9995	0.0079	21.0428	0.1663	4.1138e+03	11.9884	0.1906	5.0343e+03	11.1114	0.1414	11.9925	0.149	0.5649	1.5204e+03	16.3113
image5	0.9997	0.0067	21.7500	0.2091	1.4312e+03	16.5739	0.2322	1.2710e+03	17.0893	0.2322	1.3937e+03	16.6890	0.1950	1.3416e+03	16.8546
image6	0.9997	0.0069	21.6189	0.1156	1.3393e+03	16.8621	0.1045	1.3393e+03	16.6162	0.1070	1.4173e+03	17.1091	0.2015	1.2242e+03	17.2523
image7	0.9998	0.0027	25.7590	0.1369	1.9630e+03	15.2016	0.1278	1.9195e+03	15.2988	0.2103	15.2706	0.0970	0.2214	0.2562	14.6773
image8	0.9996	0.0084	20.7420	0.0832	1.5484e+03	16.2321	0.0769	1.7870e+03	15.6096	0.0654	1.9013e+03	15.3402	0.0791	1.7552e+03	15.6875
image9	0.9994	0.0163	17.8895	0.0723	1.9301e+03	15.2751	0.0505	2.2445e+03	14.6196	0.0439	2.0796e+03	14.9510	0.0301	2.2230e+03	14.6613

The efficiency of the proposed approach has been compared to those of well-established methods dealing with this problem in the literature. They consist in: Portilla and Simoncelli (Portilla and Simoncelli, 2000), Gatys et al (Kaspar *et al.*, 2015; Gatys *et al.*, 2015a), Deep Correlations for Texture synthesis (Sendik and Cohen-Or, 2017), c-cgCNN-Gram and c-cgCNN-Mean (Wang *et al.*, 2021), gCNN (Xie *et al.*, 2016), CoopNet (Xie *et al.*, 2018) and the Self tuning texture optimization method (Kaspar *et al.*, 2015). Figures from Fig.6.a to Fig.6.e and Fig.7.a to Fig.7.e represent respectively original images, Portilla and Simoncelli (Portilla and Simoncelli, 2000), Gatys (Gatys *et al.*, 2015a), kaspar (Kaspar *et al.*, 2015), Deep Correlations for Texture synthesis (Sendik and Cohen-Or, 2017) and the proposed results. In these experiments, test images have been taken from the CG-Textures and the Brodatz databases. All the used algorithms have been run using their default parameters as provided in their papers. The worst results have been given by Gatys algorithm (Gatys *et al.*, 2015a), followed in quality by kaspar (Kaspar *et al.*, 2015), Deep Correlations (Sendik and Cohen-Or, 2017) and the proposed results.

Table 2 shows the obtained values for SSIM, MSE, and PSNR. Carried tests have proven that the proposed texture synthesis method is adequate to represent a diversity of textures classes as it is asserted by results

of Fig.5 and Table 2. For some synthesized textures of Fig.5, PSNR values are low. However, semantic textures are preserved. They are only less clear than the original ones. For example for images 9, 7, and 5 of Fig.5, PSNR values are respectively 15.36, 17.25, and 19.42.

Our method generates as faithfully possible all textures, whether deterministic or random. Based on previous results, we can observe that the previous methods have been unsatisfactory for structures preservation. Table 3 justifies this assessment. For the four methods, Table 3 shows the obtained SSIM, MSE, and PSNR. For all tested images, the proposed method achieves the highest SSIM in [0.9994; 0.9998], the lowest MSE in [0.0027; 0.1368] and the highest PSNR in [17.8895; 25,7590], while compared methods metrics are very bad, especially regarding the MSE and SSIM values. Qualitative comparison with recent deep learning-based methods has been conducted. The obtained results are compared to the c-cgCNN-Gram (Wang *et al.*, 2021), c-cgCNN-Mean, gCNN (Xie *et al.*, 2016), CoopNet (Xie *et al.*, 2018), and the Self-tuning texture optimization method (Kaspar *et al.*, 2015).



Fig. 6: Rows 1-9 represent original and obtained images by different algorithms. Columns a-f show respectively original images, Portilla and Simoncelli, Gatys, kaspar, Deep Correlations for Texture synthesis and the proposed method to synthesize deterministic textures.

Figures from Fig. 8.a to Fig. 8.g represent respectively original images, c-cgCNN-Gram (Wang *et al.*, 2021) c-cgCNN-Mean, gCNN (Xie *et al.*, 2016), CoopNet (Xie *et al.*, 2018), Self-tuning method (Kaspar *et al.*, 2015) and the proposed algorithm.

Obtained results of gCNN (Xie *et al.*, 2016), CoopNet (Xie *et al.*, 2018), and the Self-tuning are the worst, especially on random textures. Self-tuning operates better than the gCNN (Xie *et al.*, 2016), CoopNet on the determinist ones. Both of c-cgCNN-Gram and

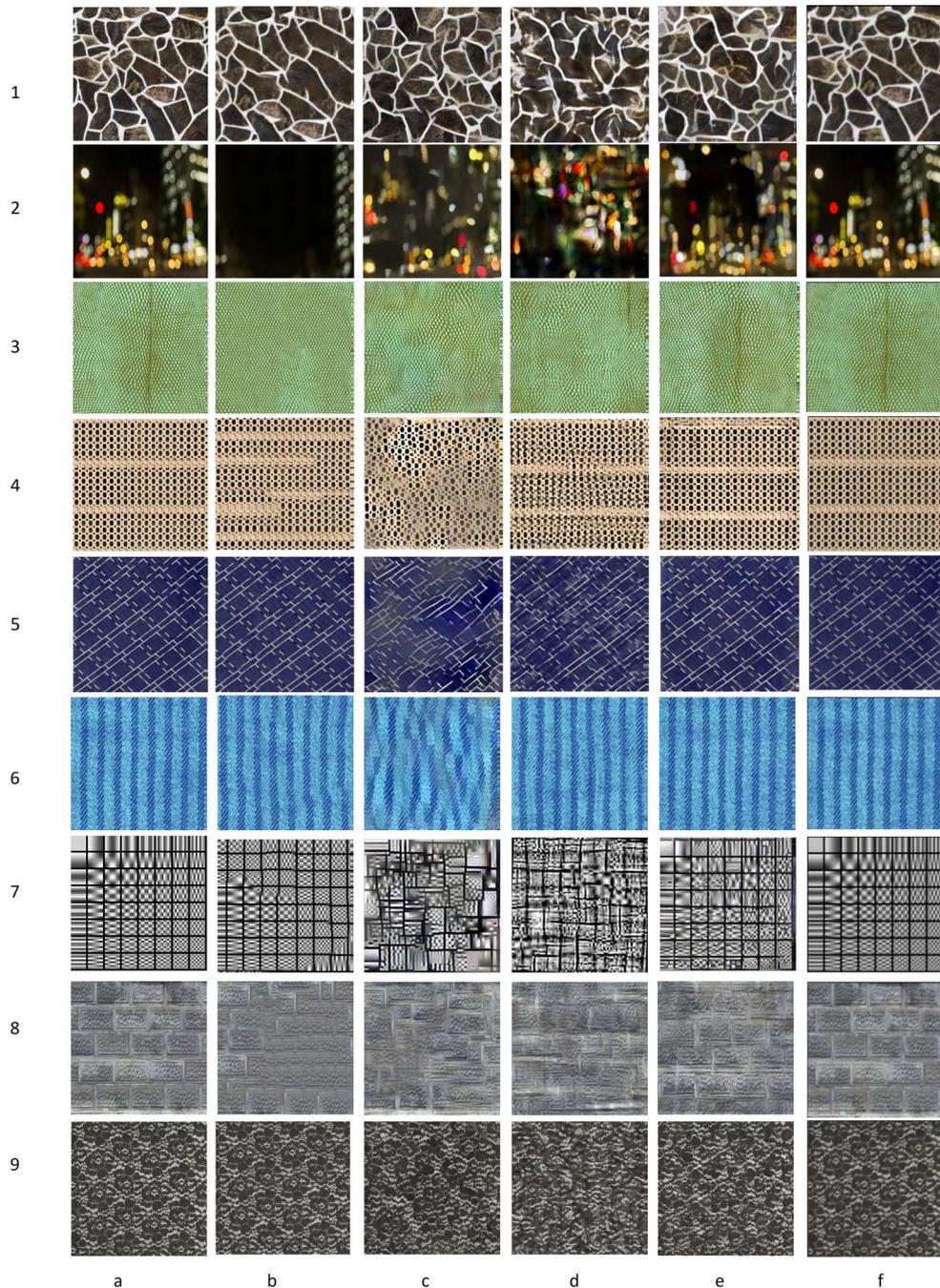


Fig. 7: Rows 1-9 represent original and obtained images by different algorithms. Columns a-f show respectively original images, Portilla and Simoncelli, Gatys, kaspar, Deep Correlations for Texture synthesis and the proposed method result

c-cgCNN-Mean produce better samples than other baseline methods. However, the produced textures are not identical to the exemplar one, unlike the proposed method, which produces textures are visually similar to the exemplar ones. And, they can also reproduce small-scale details, even for highly structured textures. We can notice that the proposed method adapts to all different deterministic and random textures compared to best methods in the literature.

It is also important to state that the proposed method's outcomes are generally comparable. We evaluate multi-scale structural similarity (SSIM) between the synthesized texture and the example. A high score indicates a high degree of visual similarity. Table 4 summarizes the quantitative results. The proposed method outperforms existing baseline methods in the majority of cases, according to the given results.

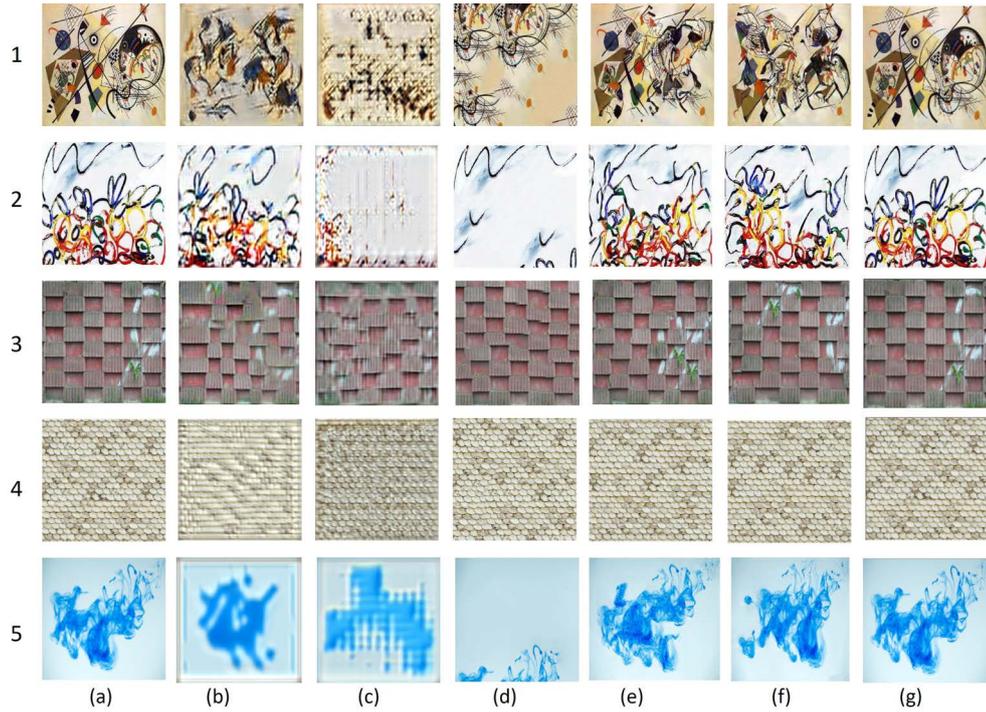


Fig. 8: Rows 1-5 represent original and obtained images by different algorithms. Columns a-g depict respectively original images, *c-cgCNN-Gram*, *c-cgCNN-Mean*, *gCNN*, *CoopNet*, *Self-tuning method* and the proposed algorithm.

Table 4: Texture synthesis result shown in FIG. 8 using SSIM.

Methods	image 1	image 2	image 3	image 4	image 5
<i>gCNN</i> Xie <i>et al.</i> (2016)	0.05	0.05	0.11	0.11	0.35
self-tuning Kaspar <i>et al.</i> (2015)	0.03	0.17	0.07	0.01	0.42
CoopNet Xie <i>et al.</i> (2018)	0.05	0.09	0.20	0.08	0.32
<i>c-cgCNN-Gram</i> Wang <i>et al.</i> (2021)	0.10	0.09	0.31	0.36	0.43
<i>c-cgCNN-Mean</i> Wang <i>et al.</i> (2021)	0.14	0.10	0.10	0.00	0.46
proposed methods	0.9099	0.927	0.9014	0.9246	0.90251

Table 5: SSIM, MSE for synthesized results of Fig.9

image	SSIM	MSE
image 1	0.9995	0.0025
image 2	0.9997	0.0023

As part of Heritage Conservation-Restoration, we tested the proposed algorithm on images representing remains from the Roman period. Fig.9.a is a Roman aqueduct and Fig.9.b is a statue of Emperor Constantin, founder of the city of Constantine in Algeria. The reconstructed images present very well quality and they are similar to the original ones. It is justified by the values of MSE and SSIM given in table 5.

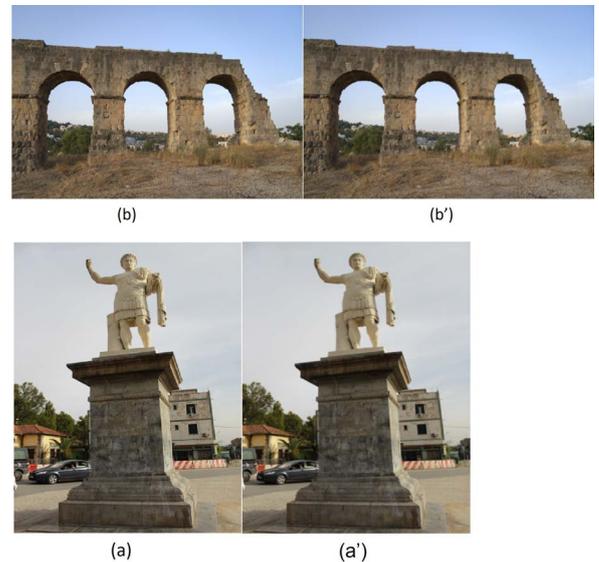


Fig. 9: Texture synthesis results using the proposed method

## CONCLUSION

In this paper, a new algorithm to reproduce and represent exemplar textures based on the 2D-RCA models has been proposed. Furthermore, approximating an image by 2D-RCA processes using

blocks has been suggested. Based on the obtained results quality, both of deterministic and random texture can be modeled by these parameters. As an advantage of this method only few parameters are needed. To evaluate the performance of the proposed algorithm, results have been compared to those given by other approaches, and the proposed algorithm's efficiency has been proved. Our approach is not complex regarding compared methods. We can notice that it adapts to all different classes of textures, unlike the other processes in state of the art deserving only certain types. Future work will consist in improving the optimisation process and implementing the proposed approach in anomaly detection context.

## REFERENCES

- Akl A (2016). Analyse/synthèse de champs de tenseurs de structure: application à la synthèse d'images et de volumes texturés. Ph.D. thesis, Université de Bordeaux.
- Basu S, Mukhopadhyay S, Karki M, DiBiano R, Ganguly S, Nemani R, Gayaka S (2018). Deep neural networks for texture classification—a theoretical analysis. *Neural Networks* 97:173–82.
- Bouleznad A, Hachouf F, Kharfouchi S (2015). Gmm estimation of 2d-rca models with applications to texture image classification. *IEEE T Image Process* 25:528–39.
- Chellappa R, Kashyap R (1985). Texture synthesis using 2-d noncausal autoregressive models. *IEEE T Acoust Speech* 33:194–203.
- Cimpoi M, Maji S, Kokkinos I, Mohamed S, Vedaldi A (2014). Describing textures in the wild. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Clark M, Bovik AC, Geisler WS (1987). Texture segmentation using gabor modulation/demodulation. *Pattern Recogn* 6:261–7.
- Cross GR, Jain AK (1983). Markov random field texture models. *IEEE T Pattern Anal* :25–39.
- Da Costa JP, Weisbecker P, Farbos B, Leyssale JM, Vignoles G, Germain C (2015). Investigating carbon materials nanostructure using image orientation statistics. *Carbon* 84:160–73.
- Duncan JS, Ayache N (2000). Medical image analysis: Progress over two decades and the challenges ahead. *IEEE T Pattern Anal* 22:85–106.
- Efros AA, Freeman WT (2001). Image quilting for texture synthesis and transfer. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.
- Galerne B, Gousseau Y, Morel JM (2010). Random phase textures: Theory and synthesis. *IEEE T image process* 20:257–67.
- Gatys LA, Ecker AS, Bethge M (2015a). A neural algorithm of artistic style. *arXiv preprint arXiv150806576*.
- Gatys LA, Ecker AS, Bethge M (2015b). Texture synthesis using convolutional neural networks. *arXiv preprint arXiv150507376*.
- Haralick RM (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE* 67:786–804.
- Haralick RM, Shanmugam K, Dinstein IH (1973). Textural features for image classification. *IEEE T systems man* :610–21.
- Hertzmann A, Jacobs CE, Oliver N, Curless B, Salesin DH (2001). Image analogies. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.
- Kaspar A, Neubert B, Lischinski D, Pauly M, Kopf J (2015). Self tuning texture optimization. In: *Comput Graph Forum*, vol. 34. Wiley Online Library.
- Kelejian HH, Prucha IR (1999). A generalized moments estimator for the autoregressive parameter in a spatial model. *Int economic review* 40:509–33.
- KHarfouchi S (2012). Statistical inference In spatial bilinear processes. Ph.D. thesis, PhD thesis, Université Mentouri Constantine.
- Lagae A, Lefebvre S, Drettakis G, Dutré P (2009). Procedural noise using sparse gabor convolution. *ACM Trans Graph TOG* 28:1–10.
- Musgrave FK, Peachey D, Perlin K, Worley S (1994). *Texturing and modeling: a procedural approach*. Academic Press Professional, Inc.
- Portilla J, Simoncelli EP (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *Int J Comput Vis* 40:49–70.
- Qian W, Cao J, Xu D, Nie R, Guan Z (2018). Synthesis of exemplar textures by self-similarity matching. *J Electronic Imaging* 27:063034.
- Reis S, Taşdemir K (2011). Identification of hazelnut fields using spectral and gabor textural features. *ISPRS J Photog* 66:652–61.
- Salmi A, Hammouche K, Macaire L (2021). Constrained feature selection for semisupervised color-texture image segmentation using spectral clustering. *J Electronic Imaging* 30:013014.
- Sendik O, Cohen-Or D (2017). Deep correlations for texture synthesis. *ACM Trans Graph ToG* 36:1–15.

- Sun W, Lu Y, Wu F, Li S, Tardif J (2009). High-dynamic-range texture compression for rendering systems of different capacities. *IEEE T Visualization* 16:57–69.
- Tong X, Zhang J, Liu L, Wang X, Guo B, Shum HY (2002). Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Trans Graph ToG* 21:665–72.
- Tuceryan M, Jain AK (1993). Texture analysis. *Handbook Pattern recogn* :235–76.
- Ulyanov D, Lebedev V, Vedaldi A, Lempitsky VS (2016). Texture networks: Feed-forward synthesis of textures and stylized images. In: *ICML*, vol. 1.
- Wang Z, Bovik AC (2002). A universal image quality index. *IEEE signal proc let* 9:81–4.
- Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004). Image quality assessment: from error visibility to structural similarity. *IEEE T Image Process* 13:600–12.
- Wang ZM, Li MH, Xia GS (2021). Conditional generative convnets for exemplar-based texture synthesis. *IEEE T Image Process* 30:2461–75.
- Wei LY, Lefebvre S, Kwatra V, Turk G (2009). State of the art in example-based texture synthesis. *Eurographics 2009 State of the Art Report EG STAR* :93–117.
- Xian W, Sangkloy P, Agrawal V, Raj A, Lu J, Fang C, Yu F, Hays J (2018). Texturegan: Controlling deep image synthesis with texture patches. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Xie J, Lu Y, Gao R, Zhu SC, Wu YN (2018). Cooperative training of descriptor and generator networks. *IEEE T Pattern Anal* 42:27–45.
- Xie J, Lu Y, Zhu SC, Wu Y (2016). A theory of generative convnet. In: *International Conference on Machine Learning*. PMLR.
- Zelinka S, Garland M (2002). Towards real-time texture synthesis with the jump map. *Rendering Techniques 2002*:13th.
- Zhang L, Zhang L, Mou X, Zhang D (2011). Fsim: A feature similarity index for image quality assessment. *IEEE T Image Process* 20:2378–86.