Interactive Texture Synthesis

Patricio Parada, Javier Ruiz-del-Solar,
Wladimir Plagges
Dept. of Electrical Engineering,
Universidad de Chile, Santiago, CHILE
Email: {pparada, jruizd}@cec.uchile.cl

Mario Köppen
Dept. of Pattern Recognition, Fraunhofer-Institut IPK Berlin, Berlín, ALEMANIA.
Email: mario.koeppen@ipk.fhg.de

Abstract

The TEXRET (TEXture RETrieval) System is a new Texture Database Retrieval System, which is based on soft-computing technologies and that is under development. One of its main features is the generation of the requested textures when they are not found in the database, which allows a continuous growing of the database. The texture generation process, implemented using *Causal Autoregressive Models* and *Interactive Genetic Algorithms*, is described in this article.

1. Introduction

This work is part of a main research effort, whose aim is the construction of the TEXRET-System, a texture retrieval system based on soft-computing technologies. The TEXRET-System has the following features: (i) direct access from the Internet, (ii) texture queries using human-like or fuzzy description of the textures, and (iii) synthesis or generation of the requested textures when these are not found in the database, which allows a growing of the database. This paper is centered in this last feature, which is implemented using *Autoregressive Generation Models* and *Interactive Genetic Algorithms* (IGA) [6]. The use of IGA requires the user participation in the generation process. It has to be noted that the suitability of the IGA is not restricted to image processing applications, but it can also be used in any *human-based* information retrieval system.

The article is structured as follows. The TEXRET-System is outlined in section 2. In section 3 is described the texture synthesis. In the section 4 is presented the interactive generation of textures. Finally, in section 5 and 6 some results and conclusions are given.

2. The TEXRET-System

The TEXRET-System, whose block diagram is shown in Figure 1, is made of the *FI* (Fuzzy Interface), the Q^2TPT (Qualitative to Quantitative Textural Properties Transformation), the *TR* (Texture Retrieval), the *TG* (Texture Generation), and the *EPA* (Evolutionary Parameter Adjustment) modules.

The on-line phase of the texture retrieval process works as follows: A human user makes a query of a texture using a subjective, linguistic or human-like texture description. The FI module enters this description into the system using a fuzzy representation of it. The Q^2TPT module interprets the query and translates it into a quantitative texture description that is implemented using Tamura Descriptors [10]. This quantitative description is used in the TR module to search the texture in the database (see description in [8]). The subjective or human-like texture description that the system accepts was determined by a psychological study described in [9].

In the case that the texture is not found in the database, i.e. the retrieved texture does not satisfy the query, the user can choose the automatic generation of it. The TG module generates the texture using an Autoregressive model [2], whose parameters are calculated from the retrieved textures. As a result of the generation process a set of textures is presented to the user. If he/she considers that one of the generated textures satisfies his query, the process finishes here. If not, the user enters into an iterative process. The iterative generation of the textures is implemented in the (EPA module). In the next two sections the modules dealing with the generation of textures are presented.

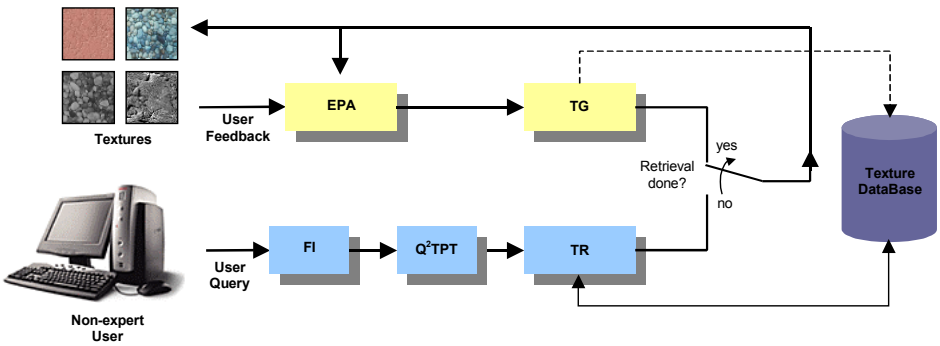

Figure 1. Block Diagram of the TEXRET System.

3. Texture Synthesis

Texture Synthesis has been an active research area in computer graphics. Among many generation methods, as for example structural and reaction-diffusion-like ones, methods considering textures as samples from probabilistic distributions are of increasing interest. By determining the form of these distributions (i.e. the model), textures can be generated. The performance of the methods depends on the structure of the probabilistic density estimator being used. In this context, Markov Random Fields (MRF) [3][4][5] and Autoregressive models [2] have been successfully used for the generation of textures.

In this work we have focused our attention to the Autoregressive models. However, at the moment we are testing the use of Markov Random Fields and high-order statistical methods to improve the generation results.

3.1. Autoregressive Model

Autoregressive models correspond to a statistical approach for synthesis of textures. They consider that the gray level $X(i)$ at the pixel position i is a linear combination of the gray levels at the neighborhood N (α_r parameters), and additive white noise that can have an arbitrary density of probability $\eta(\cdot)$.

$$X(i) = \sum_{r \in N} \alpha_r X(i+r) + \eta(i) \qquad (1)$$

The shape of the neighborhood N determines two kinds of autoregressive models. A causal neighborhood considers only the terms "before" a given pixel. A non-causal neighborhood considers both the terms "before" and "after" a given pixel. In Figure 2 these two kinds of neighborhoods are shown.

The calculation of the α_r parameters depends on the shape of the neighborhood. If it is causal, the Least Mean Square method is usually used. If the neighborhood is non-causal, the Maximum Likelihood proposed in [2] can be used.

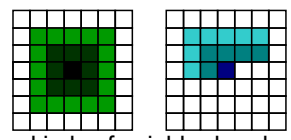

Figure 2. Two kinds of neighborhoods. Non-Causal (Left) and Causal (Right).

4. Interactive Texture Synthesis

4.1. Algorithm Philosophy

Interactive Texture Synthesis (ITS) is a dynamic process, where the user plays an important role in the synthesis of textures. This approach follows the "synthesis-by-analysis" paradigm, which means that the system analyzes a texture-sample and then it creates a new image.

In the here proposed system (see block diagram in Figure 3) a set of generated textures is presented to the user at each iteration. He selects a subset of textures that better achieve his requirements. This information is used to adjust the parameters of the texture-generation model. The system and the user iterate until the required texture is generated. What really happens is that the user performs an interactive adjustment of the texture-generation model parameters by using Interactive Genetic Algorithms [6] (IGA). IGA can be defined as genetic algorithms that do not use a standard fitness function but a user's choice as fitness criterion.

Classical texture synthesis algorithms are not universal because, they cannot be used to generate any kind of texture. If we use an adaptive approach, where the user assist to the system in the texture search, it is probable to get better results (a systems that generates a wider variety of textures). It should to be pointed out that it could be used any adaptive system that allows an user-system interaction, but is was choosen Genetic Algorithms due to their simplicity and capabilities.

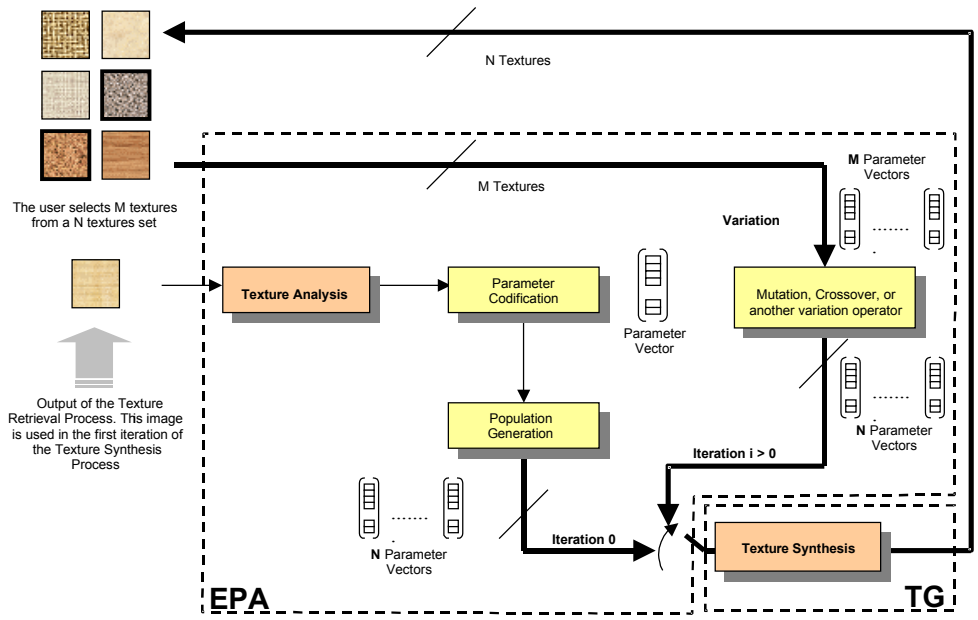

Figure 3. Block Diagram of the Interactive Texture Synthesis.

As in any genetic algorithm application, the parameters of the model (the α_r in this case) are packed into a chromosome (array), whose size depends on the texture-generation model being used. As an example, when a neighborhood of size 5 is being used, the chromosome looks like the one shown in Figure 4.

α_1	α_2	α_3	.....	α_{12}

Figure 4. The chromosome structure corresponding to the autoregressive model, with a causal neighborhood.

The texture generation process starts with the result of the retrieval process. The first vector parameters population is constructed using the parameter estimation performed by the *Texture Analysis* module (see block diagram in figure 3). Then, these parameters are coded into a vector by the *Parameter Codification* module. After that the system creates *N* chromosomes using mutation and crossover operations in the *Population Generation* module.

After these operations have been carried out, the *Texture Synthesis* module builds *N* images using the parameter vectors recently generated. Then, these images are shown to the user. In that moment, the user can interact in three ways:

- Selecting <u>one</u> of the images, if the corresponding texture satisfies her/his query,
- Selecting the <u>M textures</u> that best approach to her/his goal for to continue the iterative process, or
- <u>Rejecting all the images</u>, in the case that none of the textures look similar to her/his query.

The user selection of the textures is the only interactive part in the evolutionary procedure, and replaces the usual fitness selection. By selecting the texture images the corresponding chromosomes are selected as well. From these *M* parameter vectors, crossover and mutation operators create *N* new chromosomes. After that, *N* new texture images are generated (*Texture Synthesis* module) and presented to the user, who again has the same three options before described to interact with the system. The whole procedure is repeated until the generated textures satisfy the user wishes or after the user stop the generation process.

4.2. Implementation

Till now, the implementation of the algorithm has considered only the causal autoregressive model. For this particular model, the chromosome does not use the usual binary representation, but one where each parameter of the model is an element of the array (Figure 5). Besides that, these parameters must be restricted to the [-1,1] interval, because any value outside this interval will produce an invalid result. Taking into account this restriction, the mutation operator was implemented in a different way as usual. When the operator is applied (with a given probability), a random number is added to the parameter, but keeping the resulting value in the [-1,1] interval. The crossover operator was used in two versions: two-points and uniform, for increasing the variety of the population.

The operators are executed in a determined sequence. Firstly, mutation of the parameter vectors is applied.

Secondly, it is realized the two-points' crossover between pairs of vectors with a given probability for each pair. Finally, the uniform crossover is applied to every possible vector pair. The population size is fixed to N during the algorithm execution. For that, when mutation is performed, each of the M vectors that has been selected is used to generate $\lceil N/M \rceil$ chromosomes.

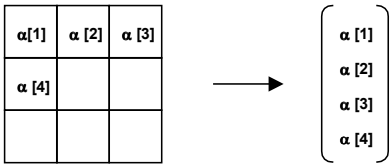

Figure 5. Chromosome Structure. The $\alpha[\cdot]$ values are the parameters of the autoregressive model (for a neighborhood of size 3).

The Interactive Texture Synthesis algorithm can be summarized as follows:

1. Calculate the autoregressive model parameters of the retrieved texture.
2. Generate a N-chromosomes population through genetic operators, using the parameters calculated in step 1.
3. Apply the autoregressive texture synthesis model (eq. 1) for generating N images, and show them to the user.
4. The user can perform any of the following actions:
   - Select one image, and quit (success!).
   - Select M images, and go to step 5.
   - Reject all the images, go to step 6.
5. Identify the parameter vectors associated to the textures that have been selected.
6. Generate a N-chromosomes population though mutation and crossover. Go to step 3.

5. Preliminary Results

5.1. Texture Synthesis

In Figure 6 we can see three examples of texture generation using the autoregressive model. In the first row of the figure are shown the original textures (from the Brodatz Album [1]), and in the second row we can observe the generated textures. In Figure 7 there are shown the generation of color textures with the same model. In both figures, the model is capable of generate structured and non-deterministic textures, but the quality of the image is directly related with the neighborhood size (15). It should be mentioned that the described texture generation methods requires a computational effort to generate a texture image that depends on the neighborhood size, ranging from a couple of seconds to a few minutes, which has an influence on the performance of the whole system.

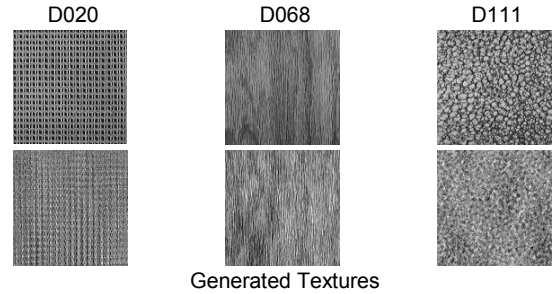

Figure 6. Textures generated using an Causal Autoregressive model (Brodatz Album).

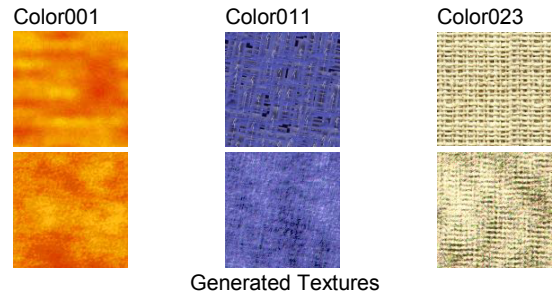

Figure 7. Textures generated using an Causal Autoregressive model. (Color Textures).

5.2. Interactive Texture Synthesis

In the Figure 8 is shown a simulation of the interactive generation process, which started with one image (Brodatz image D020), and creates two populations of textures using mutations only of the chromosomes of the selected textures (framed with a black line). As we can see, the synthesized images have emphasized particular features of the texture, like vertical lines in some textures of the first iteration, or some new characteristics, like diagonal lines in the second iteration. However, the resulting population has little variety, and the crossover operation cannot do much more. In Figure 9 we present a color texture generation example. The original image is absolutely stochastic, and the simulation process produces stochastic textures also. The interactive modification of the model parameters generates new direction of the texture pattern, but also creates some invalid images. Nevertheless, the variety of the population is not enough good. This problem motivated the use of several input textures, for achieving more variety in the generation process. In the TEXRET system these textures can be obtained from the retrieval process. In this way, not only the *top 1* retrieved texture can be considered, but also the *top j* answers to the query. In Figure 10 three textures are used as inputs to the generation process, and the results are quite good. Despite some of the images cannot be considered as textures, the majority of them present the original features combined in a new way that suggest the creation of new textures as a result. More interactive simulation results can be found in [7].

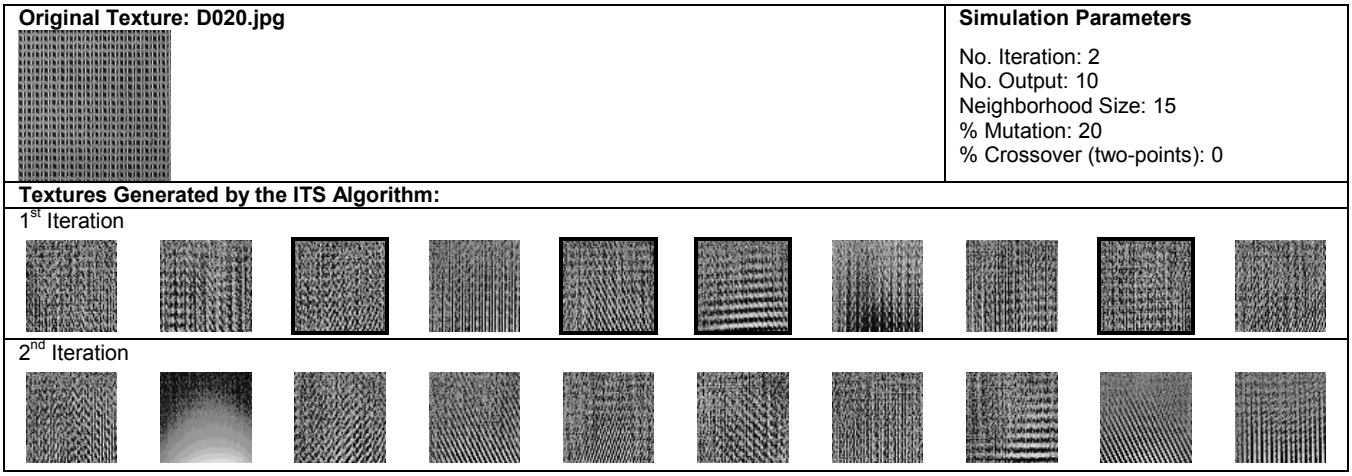

Figure 8. Simulation of ITS for Brodatz image D020.

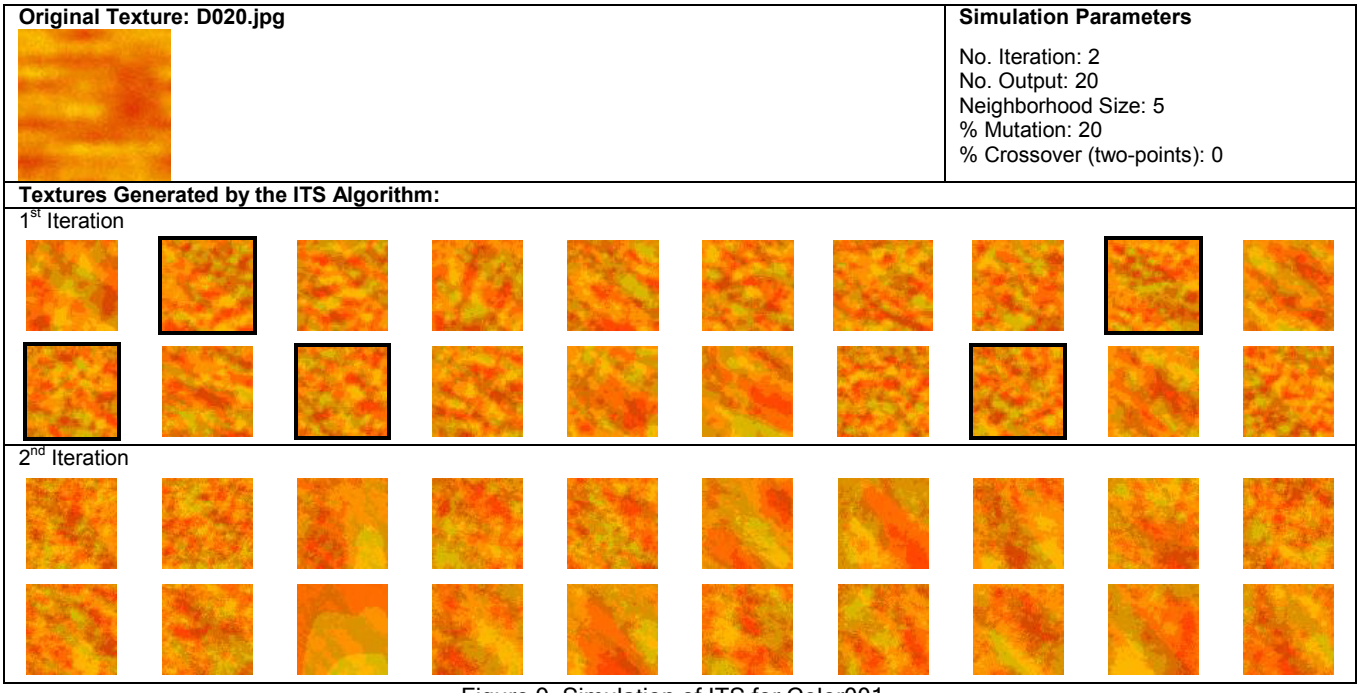

Figure 9. Simulation of ITS for Color001.

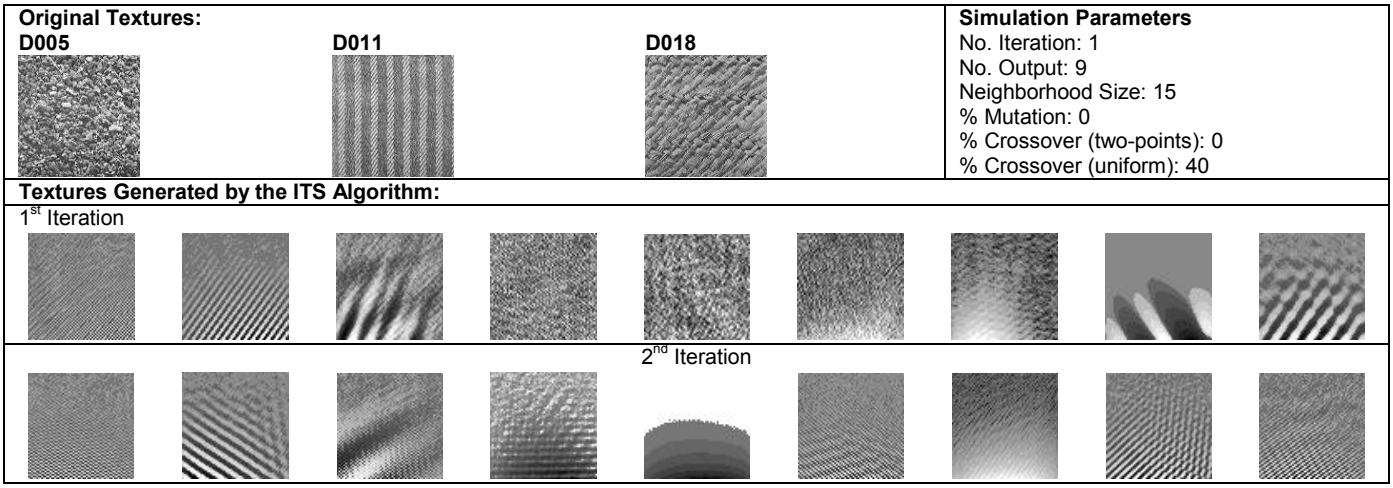

Figure 10. Simulation of ITS for Texture mixture.

5.3. Future Improvements

Until now, the ITS algorithm uses only one texture model (autoregressive). Nevertheless, one of the main features of the system is that it can work with several models (*K*). At the moment, we are working in the incorporation of MRF models to the system. In Figure 11, the block diagram of a system considering *K* different generation models is shown. This multi-model approach allows improve the performance of the whole generation process by using the advantages of each generation model. The vectors corresponding to each generation model can be understood as a different *species* in the global texture population, and because of that, it is necessary to take some precautions to combine them. In each iteration, it is generated an *N* images set, that must be divided in *K* disjoint subsets. The crossover operator can be applied only between the vector parameters of each subset (model). This schema permits the natural selection of the most suitable model for the user query. In fact, when the user chooses *M* textures from *N* images, he is also doing a selection of the texture model to be used.

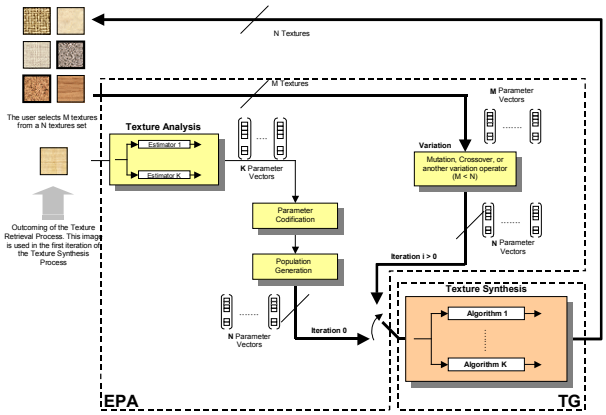

Figure 11. Block Diagram of the Interactive Texture Synthesis with K texture models.

6. Conclusions

Providing the facility of a user-friendly texture retrieval system comes out to be a very complex task. The TEXRET project corresponds to an effort in this direction. The generation of the requested textures when these are not found in the database is one of its main characteristics. In order to satisfy the requirements of the user an interactive genetic algorithm synthesizes missing textures. When the user has interactively found her/his texture, the database can be expanded and a new pair of data is supplied as well (the user-query and the texture selected by her/him). From this, even a small-at-the-beginning texture database can grow from the very beginning on.

It has to be stand out that the applicability of the Interactive Genetic Algorithms is not restricted to image processing, but it can also be used in any *human-based* information retrieval system. The preliminary results are encouraging and they suggest that the use of interactive genetic algorithms allows not only different versions of a given texture, but also create new textures in an easy way for the user. Results' improvement will be obtained by the incorporation of other textures models, as for example MRF and high order statistics models. In this moment the authors are working in this direction.

Acknowledgements

This research was supported by FONDECYT (Chile) under Project Number 1990595 and by the join "Program of Scientific Cooperation" of CONICYT (Chile) and BMBF (Germany).

References

[1] P. Brodatz, *Textures*, New York: Dover, 1966.

[2] R. Chellappa, R. Kashyap, "Texture Synthesis Using 2D-Noncausal Autoregressive Models", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP - 33, No. 1, February 1985, pp. 194 - 203.

[3] G.R. Cross and A.K. Jain, "Markov Random Field Texture Models", *IEEE Transactions on Pattern and Machine Inteligence*, Vol. PAMI- 5, January 1983, pp. 25 – 39.

[4] M. Hassner and J. Sklansky, "The Use of Markov Random Field Models for Textures", *Computer. Graphics Image Processing*, Vol. 12, April 1980, pp. 357 – 370,

[5] M. Hassner and J. Sklansky, "Markov Random Fields Models of Digitized Image Texture", *Proceedings of the International Joint Conference on Pattern Recognition*, Kyoto, Japan, November 1978, pp. 357 – 370.

[6] J. Lee, S. Cho, "Interactive Genetic Algorithm with Wavelet Coefficients for Emotional Image Retrieval", *Proceedings of IIZUKA' 98*, 1998, pp. 829 – 832.

[7] W. Plagges, "Implementation of the Evolutive Parameter Adjustment using Genetic Algorithms". Technical Report for the *Texture Synthesis Using Soft-Computing Project*, Electrical Engineering Department, Universidad de Chile, December 2000 (in Spanish).

[8] J.Ruiz-del-Solar, "Neuro-fuzzy System for the Administration of Texture Databases", *Proceedings of the XIII Chilean Congress on Electrical Engineering*, 1999 (in Spanish).

[9] J. Ruiz-del-Solar, and M. Jochmann, "On Determining Human Description of Textures". *Proceedings of the 12th Scandinavian Conference on Image Analysis – SCIA 2001*, Bergen, Norway, 2001.

[10] H. Tamura, S. Mori, and T. Yamawaki, "Textural Features Corresponding to Visual Perception", *IEEE Transactions on System, Man and Cybernetics*, Vol. SMC – 8, No. 6, July 1978, pp. 460 – 472.