Automatic Block Selection for Synthesizing Texture Images using Genetic Algorithms

*Noor Adnan Ibraheem** *Mokhtar Mohammed Hasan** *Shaima Mahdi**

Date of acceptance 15/4 / 2009

Abstract:

Texture synthesis using genetic algorithms is one way; proposed in the previous research, to synthesis texture in a fast and easy way. In genetic texture synthesis algorithms ,the chromosome consist of random blocks selected manually by the user .However ,this method of selection is highly dependent on the experience of user .Hence, wrong selection of blocks will greatly affect the synthesized texture result. In this paper a new method is suggested for selecting the blocks automatically without the participation of user .The results show that this method of selection eliminates some blending caused from the previous manual method of selection.

Key words: Texture synthesis, Patch-based, genetic algorithms.

Introduction

Texture synthesis is technique to render new texture images that are perceptually similar to the given texture samples. It has always been one of the most active research problems in computer vision, graphics, and image processing. Texture synthesis takes as input a texture of a fixed sized and produces an output texture of arbitrary dimensions which has the appearance of being made from the same source texture [1].

Currently, patch-based approach is the most effective method to solve the problem of generating textures of desired size according to a given texture and synthesizing results that should be sufficiently different from the given sample texture but should be perceived by humans to be the same texture .The main idea of patch-based approach is synthesis new texture by [1]:

1. Taking different smaller parts of given texture, i.e., patches, randomly.
2. Globally arrange patches under certain constrain.
3. Tiling them together in a consistent way.

Most of texture synthesis algorithm doses all of the three steps for one patch after the previous patch is settled.

Genetic algorithm for synthesizing texture is suggested in this work use patch-based approach to synthesize texture in a fast and easy way .The **conventional genetic algorithm uses to synthesize texture** (coined **CGTS**).In this genetic algorithm, blocks are selected manually by the user to create different chromosomes . This requires that the user possess a high level of expertise to select the better blocks. In order to use genetic texture synthesis algorithms effectively, the user must typically be a capable artist as well as having substantial technical preparation. And so, it is understandable that these

*University of Baghdad/ College of Science for Women/ Computer Science Department

requirements are frequently beyond the casual user.

This research presents an automatic approach for selecting blocks texture. The next section will give an overview of genetic algorithm. As the block is central research discussion, the next section will define it. Then, in the next section manually block selection process is described. After that fully-automatic block selection process is illustrated. Then the results form different experiments are presented .Finally, conclusions are expressed.

Genetic Algorithm: An Overview

Genetic algorithm, firstly introduced by ***Holland***[1], is an iterative procedure that consists of a constant-size population of individuals, each one represented by a finite string of symbols, known as the genome, encoding a possible solution in a given problem space. This space, referred to as the search space, comprises all possible solutions to the problem at hand.

The genetic algorithm is applied to spaces which are too large to be exhaustively searched. The symbol alphabet used is often binary, though other representations have also been used, including character-based encodings, real-valued encodings. The standard genetic algorithm proceeds as follows: an initial population of individuals is generated at random or heuristically. Every evolutionary step, known as a generation, the individuals in the current population are decoded and evaluated according to some predefined quality criterion, referred to as the fitness, or fitness function. To form a new population (the next generation), individuals are selected according to their fitness. Many selection procedures are currently in use, one of the simplest being Holland's original fitness proportionate selection, where individuals are selected with a probability proportional to their relative fitness. This ensures that the expected number of times an individual is chosen is approximately proportional to its relative performance in the population. Thus, high-fitness (``good'') individuals stand a better chance of ``reproducing'', while low-fitness ones are more likely to disappear.

Selection alone cannot introduce any new individuals into the population, i.e., it cannot find new points in the search space. These are generated by genetically-inspired operators, of which the most well known are crossover and mutation. Crossover is performed with probability *Pcross* (the ``crossover probability'' or ``crossover rate'') between two selected individuals, called parents, by exchanging parts of their genomes (i.e., encodings) to form two new individuals, called offspring; in its simplest form, substrings are exchanged after a randomly selected crossover point. This operator tends to enable the evolutionary process to move toward ``promising'' regions of the search space. The mutation operator is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space. It is carried out by flipping bits at random, with some (small) probability *Pmut*.

Genetic algorithms are stochastic iterative processes that are not guaranteed to converge; the termination condition may be specified as some fixed maximal number of generations or as the attainment of an acceptable fitness level. Below the standard genetic presented.

Genetic Algorithm in pseudo-code format.

Begin GA
 g:=0 { generation counter }
 Initialize population P(g)

[1] *J. H. Holland. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, Michigan, 1975.*

Evaluate population P(g) { i.e., compute fitness values }
While not done do
g:=g+1
Select P(g) from P(g-1)
Crossover P(g)
Mutate P(g)
Evaluate P(g)
End while
End GA

Let's consider the following simple example demonstrating the genetic algorithm's workings. The population consists of 4 individuals, which are binary-encoded strings (genomes) of length 8. The fitness value equals the number of ones in the bit string, with *Pcross*=0.7, and *Pmut*=0.001. More typical values of the population size and the genome length are in the range 50-1000. Also note that fitness computation in this case is extremely simple since no complex decoding nor evaluation is necessary. The initial (randomly generated) population might look like this:

Table 1: Randomly generated initial populating

Label	Genome	Fitness
A	00000110	2
B	11101110	6
C	00100000	1
D	00110100	3

Using fitness-proportionate selection we must choose 4 individuals (two sets of parents), with probabilities proportional to their relative fitness values. In this example, suppose that the two parent pairs are {B,D} and {B,C} (note that A did not get selected as the procedure is probabilistic). Once a pair of parents is selected, crossover is effected between them with probability *Pcross* , resulting in two offspring. If no crossover is effected (with probability *Pcross*), then the offspring are exact copies of each parent. Suppose, in the example, that crossover takes place between parents B and D at the (randomly chosen) first bit position, forming offspring E=10110100 and F=01101110, while no crossover is effected between parents B and C, forming offspring that are exact copies of B and C. Next, each offspring is subject to mutation with probability *Pmut* per bit. For example, suppose offspring E is mutated at the sixth position to form E'=10110000, offspring B is mutated at the first bit position to form B'=01101110, and offspring F and C are not mutated at all. The next generation population, created by the above operators of selection, crossover, and mutation is therefore:

Table 2: First generation calculation

Label	Genome	Fitness
E'	10110000	3
F	01101110	5
C	00100000	1
B'	01101110	5

Note that in the new population, although the best individual with fitness 6 has been lost, the average fitness has increased. Iterating this procedure, the genetic algorithm will eventually find a perfect string, i.e., with maximal fitness value of 8.

Block Definition

The genetic texture synthesis algorithm use texture patches of the input sample texture as the building blocked for constructing the synthesized texture .In each step, a block B_k (a combination of patch of pixels R_k and its neighborhoods B_k^{ov} i.e., overlap region) of the input sample texture is pasted into the synthesized texture. For simplicity, in the following figures, square blocks of a prescribed size $w_B \times w_B$ with overlap region size will use, Figure 1 gives the

meanings of block, patch and overlap region w_e. To avoid mismatching features across block boundaries B_k is carefully selected based on the blocks already pasted in output texture, from set $S_B = \{B_0,..., B_{K-1}\}$. The texture blocks are pasted in order from top to bottom, left to right. Figure 2 depicts this process. The arrangement of blocks must preserve image fidelity by obscuring block overlap regions and maintaining an element of randomness. Tiling patches with similar overlapping edges together prevents image discontinuity. In general, patch edges are concealed by blurring or redefined by a minimum–cost edge function [2,3].

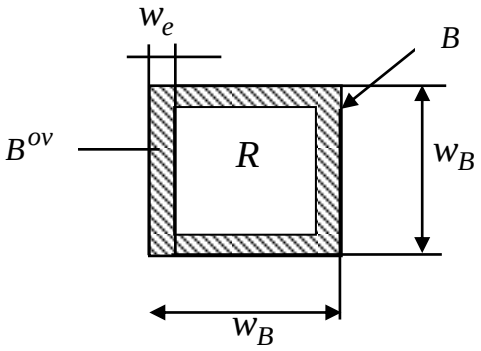


Fig.1: Block Definition. Block B consisting of patch B with surrounding neighborhood B^{ov} in the Input image.

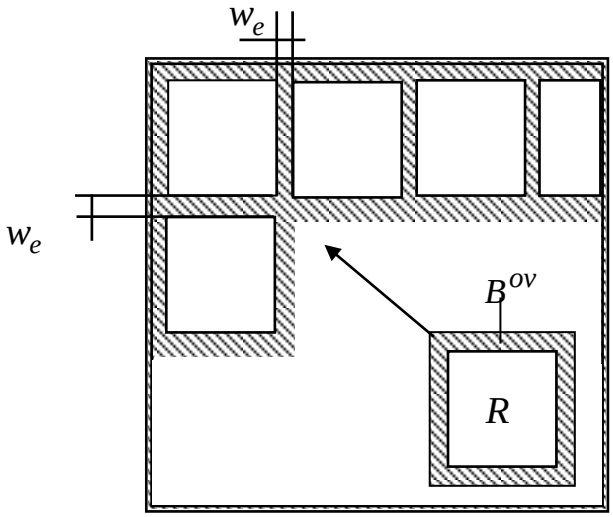


Fig.2: Arrangement of blocks in the output image.

Texture Synthesis Using Genetic Algorithm: An Overview

In this paper an algorithm is proposed for synthesizing texture using genetic algorithm. In the first algorithm, conventional genetic texture synthesis (CGTS), the conventional genetic algorithm is used. Just in the conventional genetic algorithm, works with population of chromosomes, with a generation evolution based on reproduction, with crossover. In conventional GA the usual choice of population size is based on the conception that bigger population relates to better processing, lesser chance of premature convergence, and better optimal results in [4]. Based on [5] study, the choice of population size rang from 30 to 200. The general layouts of CGTS are:

Begin

Input: Input Texture Image small size.

Output: Output texture size two times larger than Input Texture Image.

1.Generate random population of synthesized texture chromosomes with a pre-selected size.

2.Evaluate objective function and determine the best synthesized texture chromosome and carry it to the new population (elitist strategy).In this way there is a guarantee that the good synthesized texture chromosomes not lost.

3.Repeating

Select two parent chromosomes from a population according to their fitness (the better synthesized texture chromosome, the bigger chance to be selected).

With a crossover probability cross over the selected parents to form a new offspring. If no crossover was performed, offspring is an exact copy of parents.

With a mutation probability mutate new offspring at each gene.

Place new offspring in a new population. Until new population is complete.

4. Use new generated population instead of old one for a further run of algorithm.
5. If the termination criteria is not satisfied. Go to step 2.
6. Stop, and return the best solution in current population.
7. End.

Manually Block Selection Process

In genetic texture synthesis algorithms presented previously, the process of selecting blocks from the input texture is done manually .First the user selects the image of the input texture. Then he should choose the best blocks, from his perspective, to be the blocks for synthesizing texture .For simplicity square blocks are selected with a fixed size. For deterministic texture the blocks must be selected carefully to obtain output texture with good visual fidelity. On other hand, the selection of blocks for stochastic texture did not significantly affect the output texture. Figure 3 depicts this process.

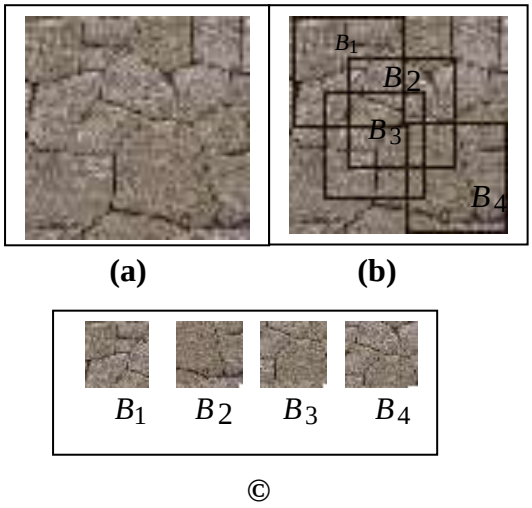


Figure 3: Selecting blocks. (a) input texture. (b) input texture with selected blocks. (c) the set S_B of selected blocks.

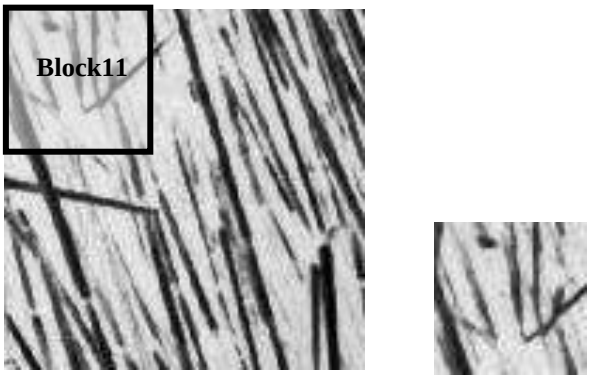
The Proposed Selection Method

Since selecting blocks manually requires that the user possess a high level of expertise. On the other hand, the proposed selection method does not depend on the user capabilities at all. It depends on the size of the overlap region B^{ov}.

Input: Input Texture Image small size.
Output: Set of block S_B .

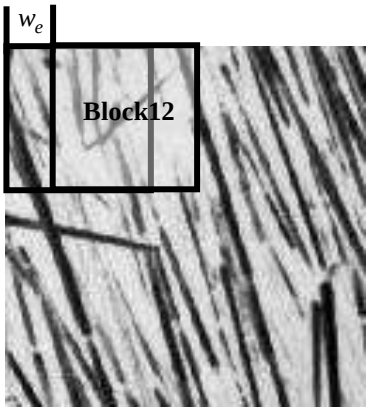
1. Selected from the most left top corner, the first block B_1 with size w_e .
2. Select the second block by shifting to the right the first block w_e pixels.
3. Continue this process until the right side of texture is reached.
4. Shift down the first selected block B_1 with size w_e pixels.
5. Go to Step2, and repeat, until all the input texture is navigated.
6. End.

Fig. (4) clarifies this process.

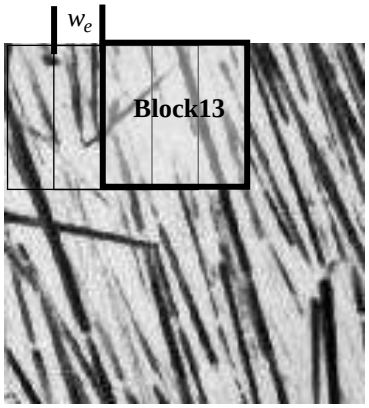


$Block_{11}$

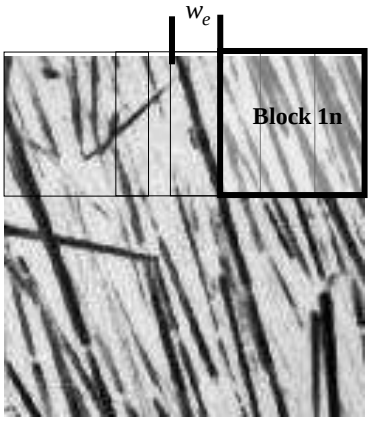
Where Block11 means, Block in row 1 and column 1 in the Input Texture Image. And so to all other images blocks.



$Block_{12}$



$Block_{13}$



$Block_{1n}$

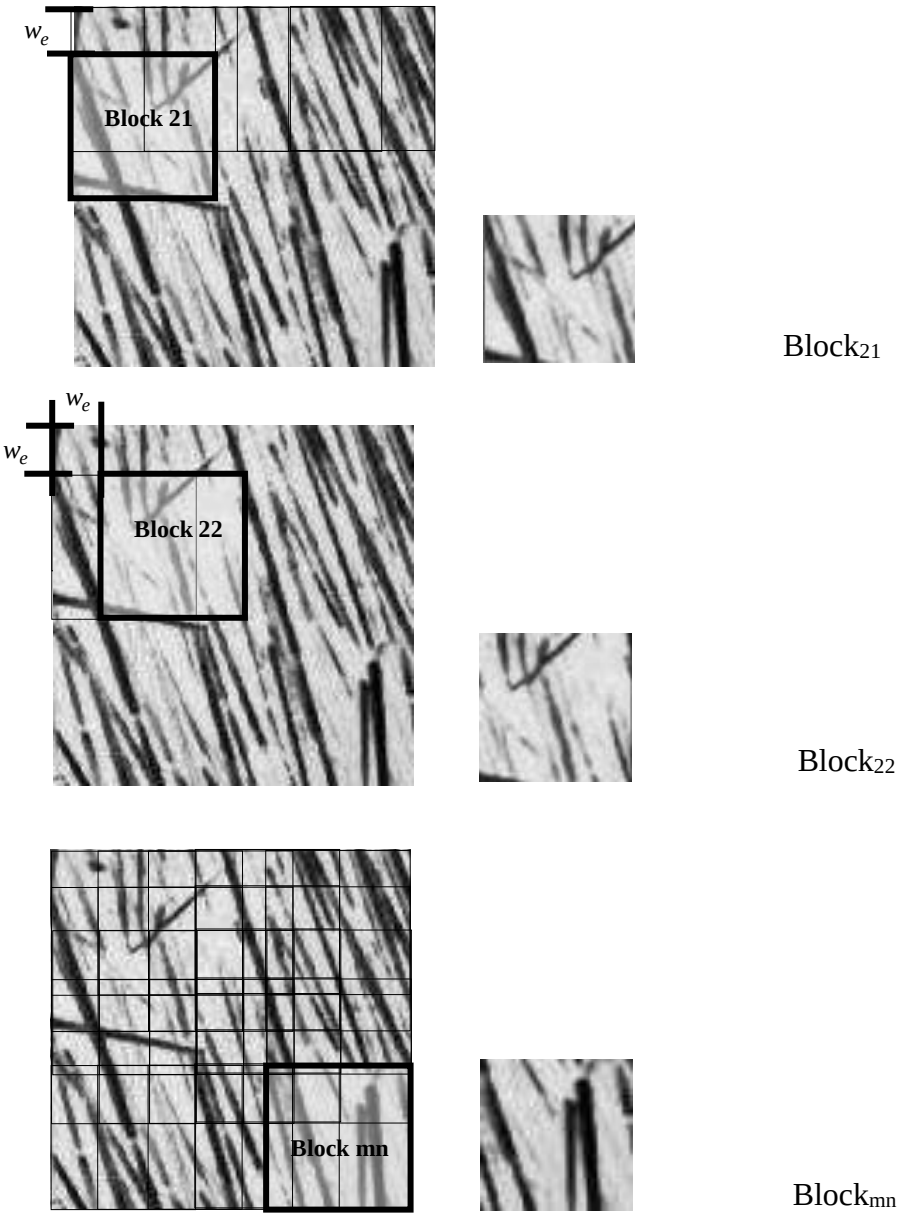


Figure 4: An example of automatic block selection.

Results

In order to emphasize the comparison between the tested texture synthesis GA using manually block selection method and automatically block selection method into a proper way, the same parameters setting are used .For selection operator, binary tournament selection is used. While two-point crossover is adopted as crossover operator with probability 0.6. Mutation occurs with probability 0.1.

The input texture size is $64*64$, while the output texture size is two times larger than input texture. The block is fixed with square shape of size $w_B = 32$ with overlap region size $w_e = 5$.

A visual comparison of CGTS with manually block selection and with automatically block selection is shown in figure 5. The input textures in this paper are from the same albums that are used to test the texture synthesis methods. These albums are Brodatz texture album [6].

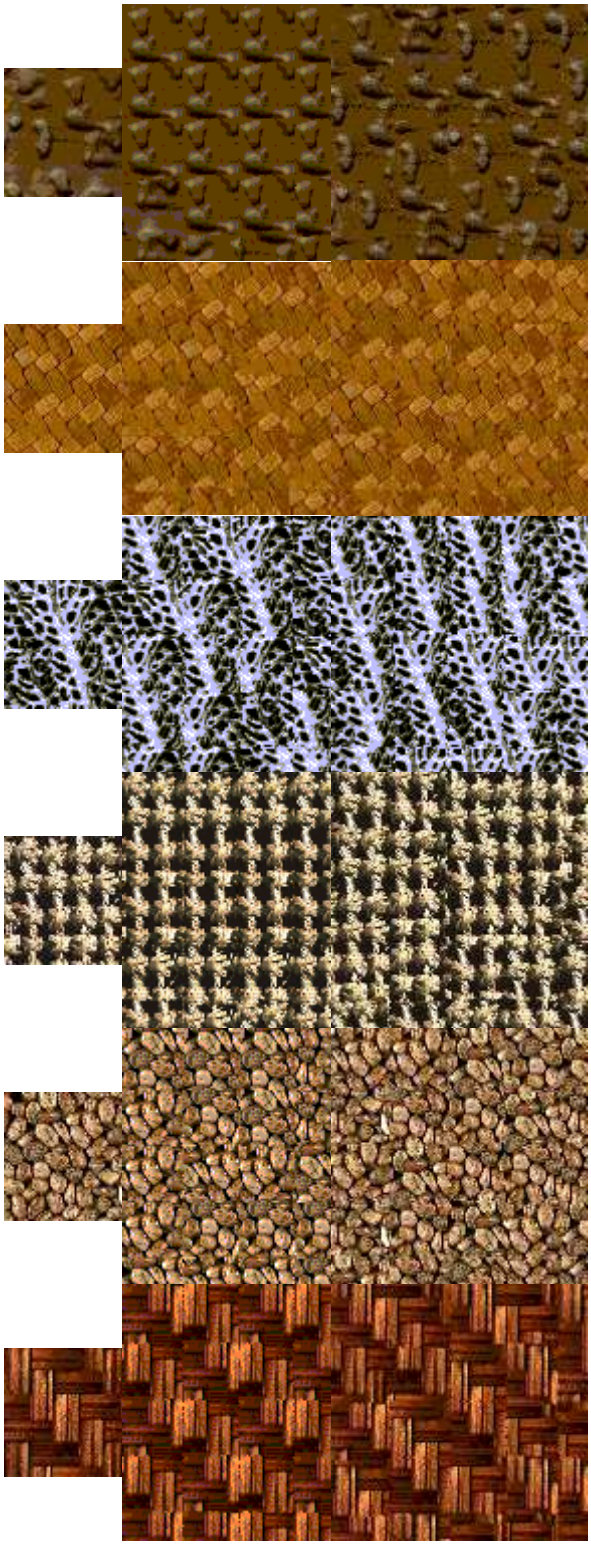


Figure 5: Results of automatic block selection for synthesizing texture using conventional genetic texture synthesis. 1st column: input texture. 2nd column: synthesized texture with automatic block selection. 3rd column: synthesized texture with manually block selection.

As depicted from the previous figures, the texture results generated using fully automatic block selection method are comparable in quality (even in some cases better quality) than those generated form using manually block selection method .In addition, there are no differences in the progress rate from applying fully automatic block selection method . Where the runtime required by CGTS rang from 20 seconds to 54 seconds per image.

Conclusions

In this paper, we have presented a new block selection method for texture synthesis using genetic algorithm. In this method, the blocks are selected automatically without any participation of user. The results demonstrate that the texture results from using automatically block selection method are comparable in both quality and computational efficiency to those generated from manually block selection method. In some cases the use of automatically block selection method eliminates some blending caused from the previous method of selection.

References:

1. **Efros, A.A., andFreeman, W. T. 2001:** Image Quilting for Texture Synthesis by Patch Based Sampling, Prints Hall, First Edition, ISBN 1-58113-292-1: 341-346.
2. **Szummer, M., and Picard, R., 1996:** Temporal Texture Modeling, Proceeding of IEEE International Conference, 3(1):823-826.
3. **Liang, L., Liu, C. Xu, Y. Q., Guo, B., and Shum, H. 2001:** Real-time Texture Synthesis by patch based, Sampling. ACMTransactionson Graphics, 20(3):127-150.
4. **Goldberg, D.E. 1989: Genetic** Algorithms in Search, Optimization,

and Machine Learning, Addison Wesley, First Edition, America: 8-12.

5. **Grefenstette, J. J. 1986:** Optimization of control parameters for genetic Algorithms, IEEE Transaction on Systems, Man, and Cybernetics, SMC 16(1):122-128.

6. **Rosalind P., Tanweer K., Fang L. 1993:** Real time Recognition with the entire Brodatz Texture Database, Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition. NewYork: 638-639.

اختيار القطع بشكل آلي في تركيب الصور النسيجية باستخدام الخوارزميات الجينية

نور عدنان ابراهيم ***مختار محمد حسن*** ***شيماء مهدي****

*جامعة بغداد/ كلية العلوم للبنات/قسم علوم الحاسبات

كلمات مفتاحيه:الصور النسيجية, نسخ مجموعة نقاط, الخوارزميات الجينية.

الخلاصة :

واحدة من الطرق لتكوين صور نسيجية بسرعة وسهولة هي باستخدام الخوارزميات الجينية. تتكون الكروموسومات في الخوارزميات الجينية المستخدمة لتكوين صور نسيجية من مجموعة عشوائية من القطع. هذه القطع المستخدمة في تكوين الصور النسيجية يكون اختيارها يدوي من قبل المستخدم . هذه الطريقة بالاختيار تعتمد بشكل كبير على خبرة المستخدم. فالاختيار الخاطى للقطع يؤثر على نتيجة تكوين الصور النسيجية .في هذا البحث اقترحنا طريقة لاختيار القطع بشكل آلي بدون تدخل المستخدم.النتائج بينت ان هذه الطريقة في الاختيار حذفت بعض عدم التجانس بين القطع المكونة للصورة النسيجية نتيجة استخدام الطريقة السابقة في الاختيار.