A GENETIC APROACH FOR AUTOMATED IMAGE GENERATION: GRAYSCALE IMAGE GENERATION

Dr.Bara'a Ali Attea Aminna Dahim Aboud
Baghdad of University \ College Of Science \
Computer Science Department

ABSTRACT

Non photorealistic rendering is a new research field in the areas of computer graphics. The goal is to give a more natural feel to computer generated images, by simulating various artistic techniques and to give the sense of an image without reproducing it. In this paper, we present a new evolutionary approach to non-photorealistic rendering of 2D black/white and grayscale images. The goal is to generate a painting that is close to a given input image. This problem can be formalized as a high-dimensional optimization problem, with local minima. We have developed a genetic algorithm that modifies the traditional uniform crossover to spread out vital genes at the expense of lethal genes rather than exchanging them between matting parents. A vital or lethal gene can be determined via a threshold field associated with each pixel gene that indicates the distance between a chromosome gene and the corresponding input image pixel. The proposed evolutionary painting framework demonstrates good results and achieves reasonable convergence.

الخلاصة

تعتبر طريقة اداء التصوير غير الحقيقي من حقول البحث الجديدة في مجال رسوم الحاسوب. الهدف هو اعطاء امكانية اكبر لتوليد الصور عن طريق الحاسوب،بواسطة محاكاة تقنيات فنية مختلفة و اعطاء مشهد الصورة بدون اعادة توليدها. في هذا البحث تم تمثيل طريقة تقدمية جديدة للتصوير غير الحقيقي للصور الأبيض و الأسود والرمادية الممثلة بشكلىمصفوفة ثنائية الأبعاد. الهدف هو توليد رسم قريب من الصور المعطاة. هذه المشكلة ممكن صياغتها من المشاكل الأمثلية المتعددة الأبعاد التي تحتوي على نهايات سفلى. تم بناء خوارزمية جينية التي تعدل التزاوج المنتظم التقليدي لنشر الجينات الحيوية على حساب الجينات الميتة بدلا"عن تبديل هذه الجينات بين الأباء المتزاوجة. يمكن تحديد الجين كونه حيوي او ميت بواسطة قيمة فاصلة مرتبطة مع كل جين مفرد و التي تعطي المسافة بين الجين للكروموسوم مع ما يقابله من قيمة مفردة في الصورة. الرسام التقدمي المفترض اثبت نتائج جيدة و التي تحقق تقارب معقول مع الصور المعطاة.

KEY WORDS
Genetic algorithms, Painter, SUX operator, and Gray scale images.

INTRODUCTION

The evolution of tools for image modeling and rendering with photorealistic fidelity-much of it represented in the twenty-six years of the SIGGRAPH conference-is a monumental achievement that has had an inestimable influence on the visual medium. Nonetheless, the tools for creating complex models require a great deal of effort and skill to use. As a result, achieving truly compelling photorealism is extremely difficult.

The goal of non photorealistic rendering is to give a more natural feel to computer generated images, by simulating various artistic techniques, such as watercolor, oil painting, or pencil sketch. The interest is also to give the sense of an image without reproducing it: digital photographs or movie sequence can be processed so they look like handmade creations. There is a real demand for fully automatic non-photorealistic rendering, especially when processing movie sequences.

Generating a painter from a 2D picture was fist introduced by P.Haeberli at a SIGGRAPH conference [Hae90]. His application introduced interactive image painting in which the user determines the number of strokes, their positions, and interactively sets the color, size and orientation of the strokes. This interactive step may require direct user input, such as the cursor location, or use an automatic input such as a gradient image. Later, P. Litwinowicz [Lit97] has made an attempt to automate these procedures in order to achieve an impressionist effect on video sequences. In his application, brush strokes are distributed over the input image, altered in random fashion, and then clipped to the edges of the image to preserve fine details. Orientation of the brush strokes is computed using the gradient direction. Optical flow field is used to move brush strokes from one frame to the next one, to avoid jitter.

Another application was that presented by P.Salisbury,P.Michael et al [Sal97] which provided an interactive system for pen and ink drawings from grayscale images. The application was not intended to be fully automatic painting as emphasis is on giving the artist a maximum control on the final result. While these applications were based on creating the painting directly from the painting in either automatic or interactive way, the work of N.Scapel [Sca] on the other hand, was based on generating several initial paintings and through evolutionary algorithm let them to evolve toward the required input picture.

In what follow, we have presented our application of image painting using genetic algorithm with new crossover operator that is a variation of uniform crossover. Section 2, presents characteristic components of genetic algorithm, how to represent genetic chromosome as an individual painting image, schemes of one-point, two-point, uniform, and SUX crossover operators, scheme of mutation operator, fitness calculation of individuals, and stopping criterion. Then, section 3 gives an outline of genetic painter algorithm. Section 4 presents results consider different parameter settings. Finally, section 5 presents conclusion and possible ramifications.

PAINTING WITH GENETIC ALGORITHM

Evolutionary Algorithms (EAs) are search techniques inspired by Darwinian's theory of evolutionary. One such evolutionary algorithm is the Genetic Algorithm (GA), which has proven to be very successful in finding good solutions for difficult real world problems. In the rest of this section we present genetic algorithm together with its characteristic components as used to evolve paintings. For more information on genetic algorithm one can refer to [Gol89][Mit96][Mic99].

In a GA, a population of candidate solutions is maintained. Each individual in the population is represented by a chromosome, which encodes some candidate solution to the problem at hand. Additionally, each individual is assigned a fitness score, which measure the quality of the solution encoded by the individual. These fitness scores are used to probabilistically select individuals for reproduction, in the sense that fitter individuals will be more likely to be selected than those less fit. This parallels Darwinian evolution's" survival of the fittest", in that fit individuals survive and procreate, while weak individuals die off and become extinct.

Representation scheme

The genetic material of the individual is stored in a chromosome made up of basic genetic building blocks called genes which shape the physical features of the individual. A chromosome with n genes is shown **Fig. (1)**.

G_1	G_2	G_3	....	G_n

Fig. (1) A chromosome of n-genes.

In genetic algorithm, a population means a set of chromosome stored, generally, in binary strings. However, in our case, we have defined an individual as a sequence of two-field records, each record has a gray level, G, and a flag, F. Gray levels are defined in the range [0, 63] interval. The flag, F, is either 0 or 1. Our implementation is based on fixed length GA's, where the total number of genes in a chromosome is equal to the size of the input image. An individual example (for input image of size 64*64) is shown below **Fig. (2)**.

F	G	F	G	...	...	F	G
0	11	1	32	...	...	1	7
Pixel(1,1) Gene-1-		Pixel(1,2) Gene-2-				Pixel(64,64) Gene-4096-	

Fig. (2). An individual example.

The above chromosome can be expressed as a 2D image of size 64*64 by a painting program which reads G parameters and apply them in the canvas. The flag value 0(e.g., in gene-1-) tell the painter that the gray level at this location is far, in some metric, from the original input image pixel. While flag equal to 1 means that the corresponding chromosome pixel is get close to that of the input image pixel at the same location. Note that a chromosome representation for binary image does not require F fields.

The genetic cycle

Genetic algorithms consist of three basic operators: selection, crossover, and mutation that shape populations. A GA most commonly starts with a randomly produced initial set of chromosomes subject to some constraints. In our case, the gray value, G, field of genes of a chromosome are randomly set with range 0-63, while flag fields are then determined either to be 0 or 1 according to a preselected distance difference selected by the user. The distance difference value determines the number of successive gray levels that can be considered as a single level from the user point of view.

Once we have an initial set of chromosomes, the genetic cycle starts. From each chromosome in the set, the gene gray values of the individual are extracted and the fitness of the individuals is evaluated using the criterion function (to be presented later). When the fitness's of all chromosomes have been evaluated we can form the mating pool which is a collection of individuals who will have the right to reproduce themselves into future generations.

Selection

The selection of individuals that form mating pool is usually, in genetic algorithm, random, not deterministic. However, each individual has a probability proportional to its fitness of getting into the mating pool. There are a number of schemes for performing selection. The most commonly used is tournament selection. Here, a set of two or more individuals are randomly selected from the population. Then the individual with the highest fitness score is retained as a parent in the mating pool for reproduction.

Crossover operation

Crossovers, as well as mutation, appear to be the perturbation operators in natural evolution. They change the participating sequences and thereby generate new samples. In GAs, emphasis in mainly concentrated on crossover, the recombination operator of GA, as the main variation and exploration operator that work by swapping portions between two individuals [Gol89]. There are various crossover operators, but all aim at recombining the genetic material of the two parents in an efficient manner. The most common approach is to think of chromosomes as one dimensional strings, to divide them into several pieces(usually two, so as to say two-point crossover)and swap the pieces from the two parents to form two new offspring chromosomes. However our approach considers the chromosome as two dimensional formations and carries out the swapping accordingly. An important parameter of GA crossover is crossover probability; Pc The normal range for this value is between 0.5 and 0.8.

The Spread out Uniform Crossover (SUX)

Here, we have modified the uniform crossover operator to be as what we coined it Spread-out Uniform crossover , SUX. In SUX, the genes between two parents are spread-out rather than swapped. If the flag field of a gene of one parent is 1, and the corresponding flag field in the gene in the second parent is 0, then the gene gray value of the first parent is copied into the gray value of the gene of the childs. If both thre flag are 0's, then swap them with probability 0.5 to the childs. In case of both flags are 1's, then copy to childs the gray level value of the parent that is more close to the original pixel value of the input image (See **Fig. (3)**).

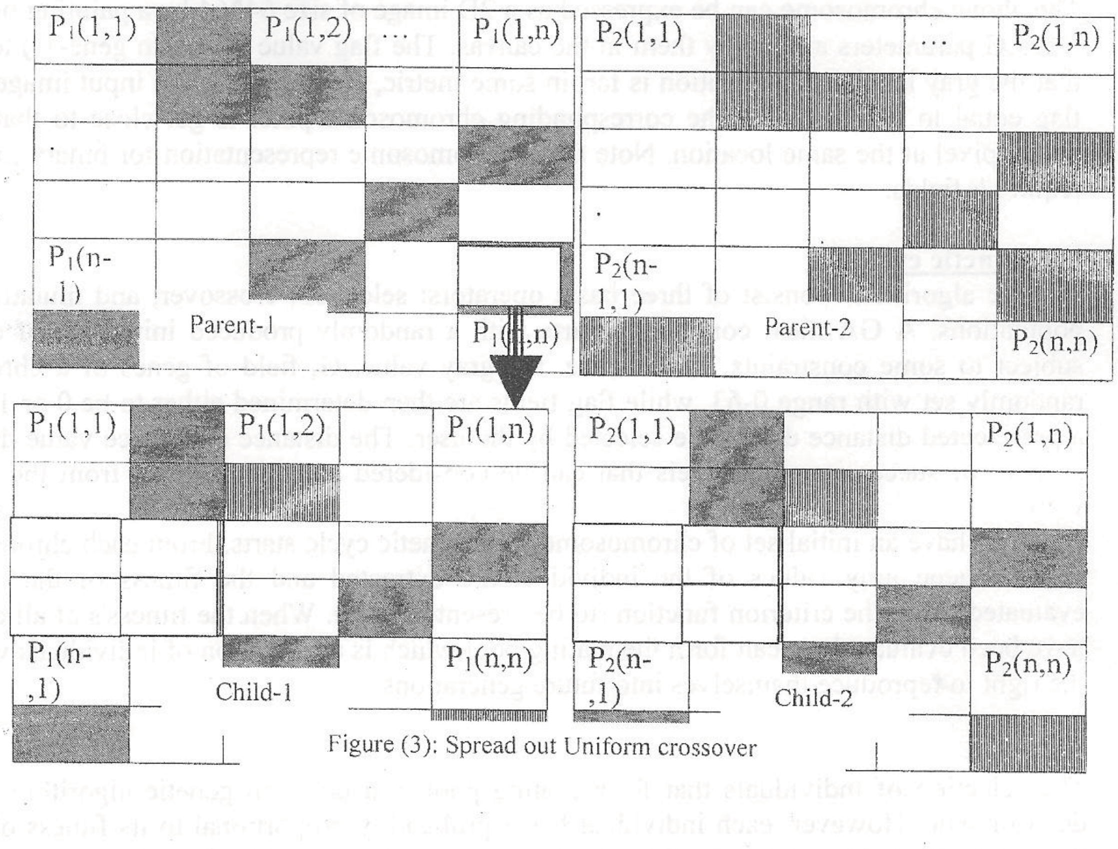

Figure (3): Spread out Uniform crossover

mutation

The second reproduction operator that manipulates the genetic material is mutation which randomly alters the genes (here G fields) of a population with a predetermined, usually, low probability P_m. It serves two purposes: to introduce new genetic structures to the genetic cycle and to regenerate lost genetic structure. Moreover, mutation can regenerate structures that become extinct during the genetic cycle to conserve the diversity of the search.

Fitness measure

Let I be the input image. For each individual, an image P is constructed which is of the same size of I. The fitness of each individual is computed as distance between P and I:

Fitness (P)=dist (P, I) .(1)

Where

$$\text{Dist}(P, I)=\sum_{i=1}^{x}\sum_{j=1}^{y}\left|P(i,j)-I(i,j)\right| \quad (2)$$

Where x, and y represent respectively height and width of the input image.

As this fitness function represents the distance difference between input image and ally generated image, we have to seek for those individuals in which this distance metric is as small as possible.

Stopping criterion

The GA cycle never terminates in nature; however we do not want an optimization algorithm that executes forever, therefore, we have to set a stopping criterion for the GA cycle. The stopping criterion used is the maximum number of generations the genetic cycle is allowed to run. A small stopping criterion may not give the GA enough time to reach an optimum. On the other hand, a very large stopping criterion is unnecessary because there can be no further gain once the optimum solution is reached.

THE EVOLUTIONARY PAINTING FRAMEWORK

Given the mechanisms of genetic algorithm operators (selection, crossover, and mutation) together with individual data structure fitness evaluation, and stopping condition, the framework of evolutionary painting can be described. Algorithm below outlines in pseudo codes the steps of evolutionary painting.

Algorithm: Evolutionary Painting

```
Input: black/white or Grayscale-Image
Output: Evolutionary painting Image
  1. [Start] Generate random initial population of images with sizes equal size of input image.
  2. [Evaluate] Evaluate the fitness of each chromosome in the population.
  3. [Generate] Create a new generation of individuals through :
       3.1 . [Select] Create a mating pool of possibly good parents from old population via tournament `selection with predefined tournament size.
       3.2 . [Crossover] pick two randomly parents from gene pool and crosses, with probability $P_c$, them to create two new childs.
       3:3 . [Mutate] Mutate each generated child's gene with probability $P_m$.
  4. [Replace] use new population instead of old one for further run.
  5. [Stop] end the run if the stopping condition is satisfied.
```

RESULTS

Synthetic Tableau

Several instance for the GAs and images are consider as illustrated in Table (1)

Table (1) The most argument used in GA.

GA No.	Input Image	Pop_size	Toyrnament_size	Crossover_type	Stop_condition	Masking
GA_1	Black/White Plus Shape	200	2	Two-point	300	No
GA_2			4			
GA_3			8			
GA_4						Yes,size 3*3
GA_5	64-grayscal cat		2	One-point	150	No
GA_6			4			
GA_7			8			
GA_8			2	Two-point		
GA_9			4			
GA_{10}			8			
GA_{11}			2	Uniform		
GA_{12}			4			
GA_{13}			8	Uniform		
GA_{14}			2	SUX operator		
GA_{15}			4			
GA_{16}			8			

Quality Measure

In fact, there is no good objective criterion available for measuring the perceived image similarity. However, there are a number of common error measurements. Mean Square Error (MSE) measures the average amount of difference between pixels of an image and GA individual image. If the MSE is small, the GA individual image closely resembled the original. Below is the formula of MSE calculation:

$$MSE = \frac{\sum_{i=1}^{X}\sum_{j=1}^{Y}|I(i,j) - P(i,j)|}{XY}$$

Where

I(i, j) : The original image.
P(i, j) : The GA individual gene.
X, Y: Dimension of the image.

Experimental Result

Fig. (4) presents results of GA$_1$, GA$_2$, GA$_3$, and GA$_4$ for black/white plus-shaped image.

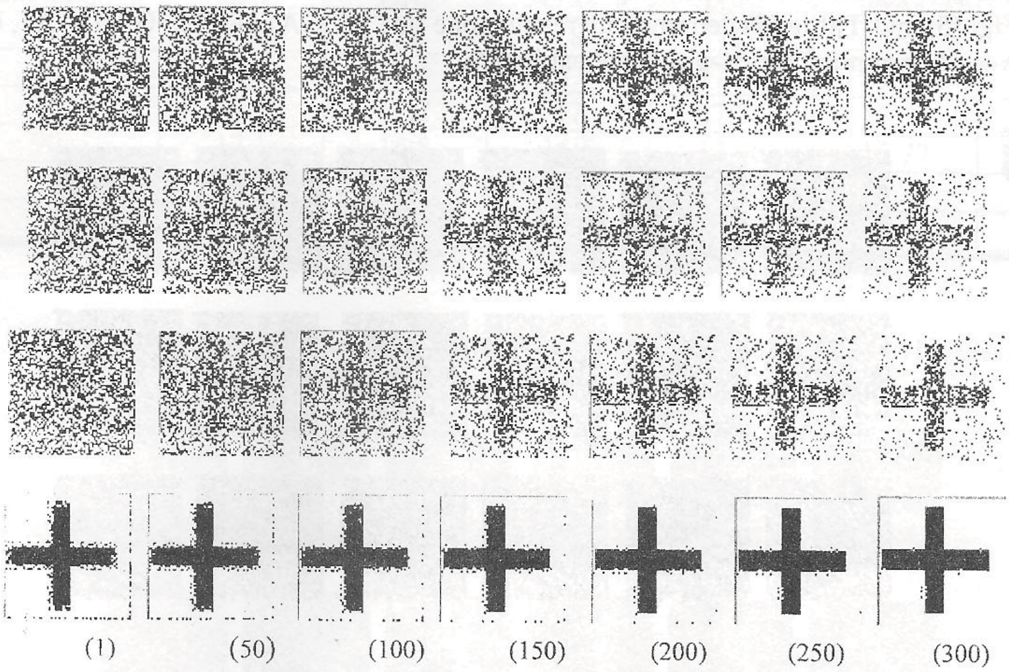

Fig. (4) Represent output images at generation (1, 50,100,150,200,250,300)
by using respectively GA$_1$, GA$_2$, GA$_3$ and GA$_4$.

Table (2) MSE for respectively GA$_1$, GA$_2$, GA$_3$, and GA$_4$

No. of generation	MSE of GA$_1$	MSE of GA$_2$	MSE of GA$_3$	MSE of GA$_4$
1	0.476	0.476	0.476	50.062
50	0.395	0.398	0.394	0.04516
100	0.355	0.348	0.336	0.03125
150	0.318	0.302	0.283	0.02124
200	0.286	0.261	0.241	0.0122
250	0.255	0.227	0.205	0.00659
300	0.228	0.196	0.174	0.0034

Our first step in experiment was to test the genetic algorithm simply for binary image painting. The initial results support the view that traditional Gas needs some alternative strategy for its operators (e.g., crossover) to deal effectively with this state. As shown from the results, varying size of tournament does not significantly effect on the performance of the GA. Further, we feel that results above set the stage for the design of other crossover operator that can be exploited well in Gas for painting images with larger search space (e.g., grayscale image) than binary ones. Also, from the results above, one can see that the resulted evolved image provided by GA1, GA2 and GA3 contains random noise distributed over the image.

Here, we overcome this by denoising it using a masking with preselected size that moves throughout image, determines the existence of noise at this mask, and remove it. Results of GA_4 in **Fig. (7)** depict results of this denoising method.
Next. **Fig. (5)** and **Table (3)** demonstrate results (one line per GA) of GA_5, GA6, GA7, GA8, ĞA9, and GA_{10} when applied for 64-grayscale cat image.

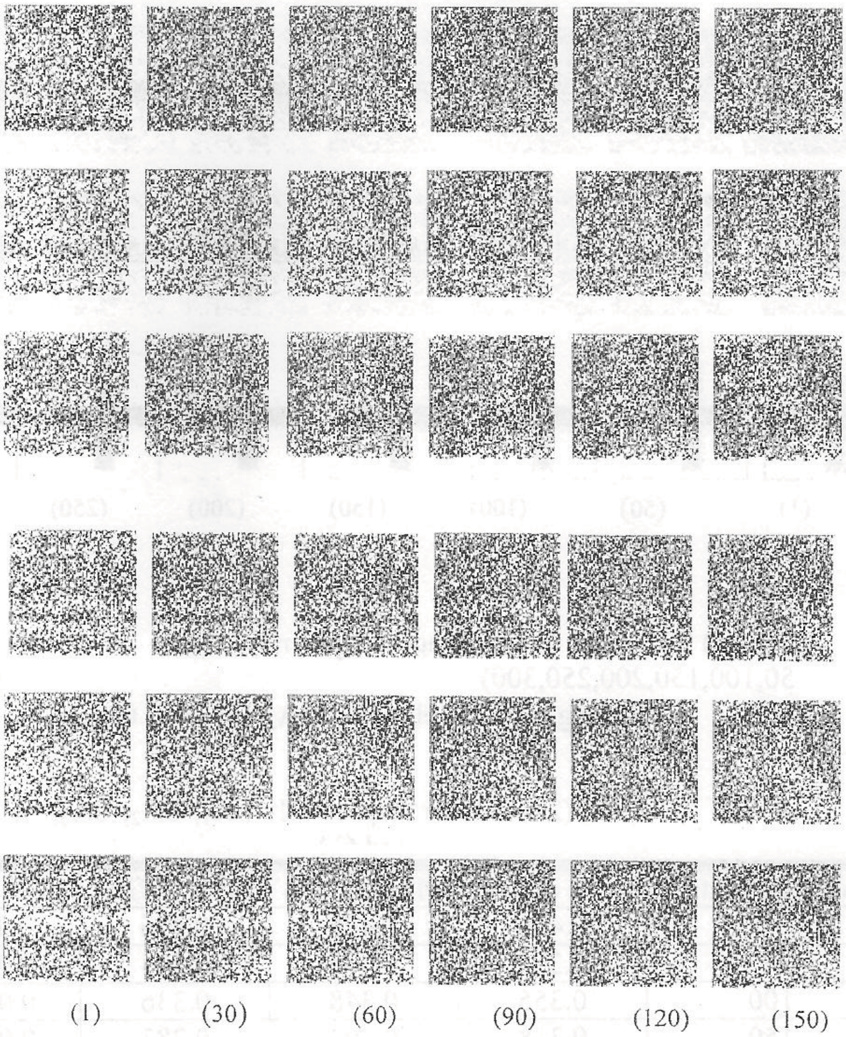

Fig. (5): Represent output images at generation (1, 30, 60, 90,120,150)
by using respectively GA_5, GA_6, GA_7 , GA_8, $GA_9$and GA_{10}.

No. of generation	MSE of GA_5	MSE of GA_6	MSE of GA_7	MSE of GA_8	MSE of GA_9	MSE of GA_{10}
1	18	18	18	18	18	18
30	17.7	17.6	17.5	17.65	17.55	17.45
60	17.4	17.2	17	17.36	17.12	16.94
90	17.1	16.8	16.5	17.06	16.77	16.50
120	16.8	16.4	16	16.78	16.38	16.02
150	16.5	16	15.6	16.52	16.03	15.62

Table (3) MSE for respectively GA_5, GA_6, GA_7, GA_8, GA_9, GA_{10}.

Finally. Fig (6) and Table (4) demonstrate results (one line per GA) of GA_{11}, GA12, GA13, GA14, GA15, and GA_{16} when applied for 64-grayscale cat image.

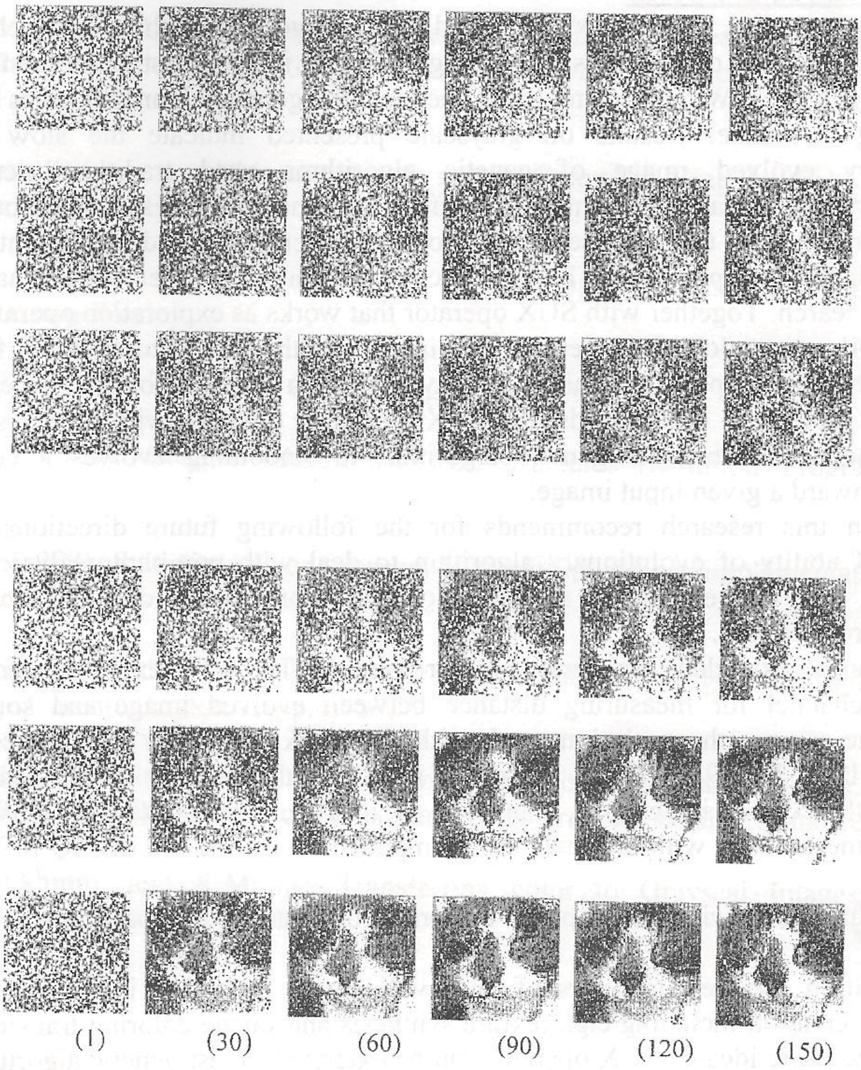

Fig. (6): Represent output images at generation (1, 30, 60, 90,120,150)
by using respectively GA_{11}, GA_{12}, GA_{13} , GA_{14}, GA_{15} and GA_{16}.

Table (3) MSE for respectively GA$_5$, GA$_6$, GA$_7$, GA$_8$, GA$_9$, and GA10.

No. of generation	MSE of GA$_{11}$	MSE of GA$_{12}$	MSE of GA$_{13}$	MSE of GA$_{14}$	MSE of GA$_{15}$	MSE of GA$_{16}$
1	18	18	18	18	18	18
30	15.6	15.3	14.9	12.7	9.1	5.4
60	14.2	14.14	13.8	10.5	4.7	1.9
90	13.3	13.4	13	7.7	2.4	1.3
120	12.7	12.7	12.3	6.1	1.5	1.1
150	12.2	12	11.7	5.2	1.2	1

Conclusion and future works

In the previous section, we have experimented genetic algorithm with different characteristic setting to evolve a population of paintings toward a given input image. First experiments were applied for 64*64 binary images. Without using SUX operator, the genetic algorithm gives satisfactory results but at long run time. Results on grayscale presented indicate the slow convergence and unsatisfactory evolved image of genetic algorithms used traditional crossover schemes (one-point, two-point, and uniform) with different tournament sizes. On the other, the results of genetic algorithm with the proposed SUX crossover operator provide sufficient paintings that are close enough to the input image. Further, the results show the effect of tournaments size on the speed of the search. Together with SUX operator that works as exploration operator, the tournament selection works as exploitation operator, and increasing the size of tournament from 2 to 8 further speeds up the search process achieving convergence in less number of generations (see result presented in tables). In other words, the SUX crossover operator with proper size of tournament selection operator enable the genetic algorithm to smoothing evolves a completely random population toward a given input image.

The work in this research recommends for the following future directions. Result gives an indication of ability of evolutionary algorithm to deal with non-photorealistic rendering (NPR) problems. It can be generalized to include water-color, paster, oil, and texture modification (editing and restoration).

The work can be extended to generate a color painting. The work experimented with images that used RGB channel for measuring distance between evolved image and source input image. However, one can use the evolutionary algorithm to work with other color space (e.g., LαB, YIQ, YUV , or HSV) [Run98][Tas01], to search for optimal lightness values near the given input lightness of the colored image (i.e., L in LαB, Y in YIQ, Y in YUV, H in HSV) , combine these evolved lightness image with original color components (i.e., αB in LαB, IQ in YIQ, UV in YUV, SV in HSV).

The genetic algorithm with the proposed uniform crossover"SUX" operator gives us an interesting and a

new ammunition for the supporters of this evolutionary algorithm in other fields of automatic artistic styles creation including e.g., texture synthesis and image coloring transfer between images problems. The basic idea of SUX operator can be extended to test genetic algorithm to constrained texture synthesis problem in which a scrambled region, scratches, wires or props in an image to be replaced by synthesized texture that must blend seamlessly with the existing texture of the input image [Wei01]. The threshold fields here can give an indication about two factors that must be satisfied: the synthesized genes must look like the surrounding texture gene of the image, and the boundary between the new and old regions must be invisible. Additionally, the idea of SUX operator in genetic algorithm can be exploited in color transfer between images problem. In color transfer between images, the problem lies transferring color “mod” of a source to a target image by matching luminance and texture information between these two images [Wel]. Then, the threshold field of genes can give an indication about matching in luminance (both average and neighbor hood statistics) between target gene pixels and best selected source pixels.

REFERENCES

P. Haeberli. (1990), Painting by Numbers: Abstract Image Representations, SIGGRAPH 90 Conference Proceedings, Annual Conference Series, PP.207-214. ACM SIGGRAPH, ACM press, August.

P.Litwinowicz. (1997). Processing Images and Video for An Impressionist Effect”, SIGGRAPH 97 Conference Proceedings, Annual Conference Series, PP 407-414. ACM SIGGRAPH, ACM Press. August.

P.Salisbury. P. Michael etal. (1997). Orientable Textures for Image-Based pen-and-Ink Illustration, SIGGRAPH 97 Conference proceedings, Annual Conference series, PP. 401-405. ACM SIGGraph.ACM press, August.

N. Scapel, Genetic Painter, available at research report, Computer Graphics Lab, Department of Computer Sience, Stanford University.

D.E. Goldberg, (1989), Genetic Algorithms in Search, Optimization, and Machine learning, Reading, MA:Addison-Wesley,.

M.Mitchell, (1998), An Introduction to Genetic Algorithms”, MIT Press,.

Z. Michalewicz. (1999). Genetic Algorithms+Data structures=Evolution Programs, Springer-Verlag,.

D.L. Ruderman, (1998), Statistics of cone responses to natural images implementations for visual coding”. J.Opt.Soc.Am.A/Vol.15,No.8/August pp.2036-2045,.

T.Tasdizen,L.Akarun, and C.Ersoy, (1998), Color Quantization with Ggenetic Algorithms, Signal Processing: Image Communication 12,pp.49-57,.

Li-Yi Wei. (2001), Texture Synthesis By Fixed Neighborhood Searching, Ph.D.dissertation, Department of Electrical Engineering and the committee on graduate studies, Standford University,.

T.Welsh. M.Ashikhmin, and K.Muller, Transferring color to Grayscal Images, available at http://www.cs.sunysb.edu/~tfwelsh/ colorize.