

## AN EVOLUTIONARY BASED IMAGE GENERATOR

Amenah Dahim Abooud

Computer Science Department, College of Science, University of Baghdad.

### Abstract

This paper presents a fast algorithm that utilizes evolutionary algorithm for patch based image generation. The algorithm attempts to find the most appropriate collection of source square image patches that can generate altogether a given colored/gray target image. Each patch holds complete information for both luminance and chromatic components from the source image to be used to generate a specified target patch. The difficulties are occurred when chromatic and luminance information is to be transferred to a blank canvas to generate a specified target image. The proposed Evolutionary Algorithm (EA) is used to overcome these difficulties and attempts to find an acceptable and perceived solution to this problem by the search for best patch luminance matching until a specified number of generations is met. Also, this algorithm modifies the traditional uniform crossover to a multi-sexual recombination operator which inherits vital genes among all individual to one super generated offspring. All other offspring are generated by mutating that super offspring. The experimental results have demonstrated that good results can be obtained with reasonable convergence speed. Also, results show that when the patch size is increasing, the mean square error is decreasing.

**Keywords:** Image Generation, luminance and chromatic, evolutionary algorithm.

### 1. Introduction

Image Generation is one of the important image processing techniques which makes color modification process in a given image simple and gives visual appeal for some images. Color transferring process from one image to another is inspired by work of Welsh et al [2]. In their work, colors from a source image are transferred to a target gray-scaled image using a simple procedure. Their basic method matches the one-dimensional distribution of luminance values. Transferring only one component (luminance channel) simplifies this problem to the generation of gray-scale images. The task of "generating" color image involves assigning three-dimensional (RGB) pixel values to an image. The characteristics generally used to distinguish one color from another are brightness, hue and saturation. Brightness refers to intensity, hue is an attribute associated with the dominant wavelength in a mixture of light waves, saturation refers to relative purity or the amount of white light mixed with a hue. The Hue and saturation taken together are called chromaticity, and therefore a color may be characterized by its brightness and chromaticity. Since different colors may have the same luminance value but

vary in hue or saturation, the problem of generating color images has no inherently "correct" solution. Further, color can be added to a range of scientific images for illustrative and educational purposes [1][2][3].

The idea of image generation is inspired by image colorization which is in general a severely under-constrained and ambiguous problem for which it makes no sense to try to find an "optimum" solution, and for which even the obtainment of "reasonable" solution requires some combination of strong prior knowledge about the scene depicted and decisive human intervention. Even in the case of pseudo generating, where the mapping of target luminance and chromatic values to source luminance and chromatic values is automatic, the choice of this map is commonly determined by human decision [4].

There are many works in the field of transferring color components between images, some of these are presented here briefly. Welsh et. al.[2], introduced a general technique for colorizing gray scale images by transferring color between a source color image and a target gray scale image. This approach work by comparing luminance values of the neighborhood for each pixel in the target image with all neighborhoods for

each pixel in the source image to find the best pixel and add its chromatic value to the luminance value of the target image. Also, Di Blasi and Reforgiato[3] improves the Welsh et al[2] work by adopting an efficient data structure for fast colors retrieving. Ammar A. and Aminna [5] introduced a technique for generating images by comparison each pixel with its neighborhoods in a target image with all pixels in a source image to determine the best pixel and add its luminance and chromatic values to target image. This technique requires long time to find the required solution. Bara'a A. and Aminna [4] proposed an evolutionary colorizing algorithm that uses first- and second- order statistics for luminance and texture matching between the target and source patches. Further, an interesting thing happened recently, a Mona Lisa replica was made of 3,604 cups of coffee - and 564 pints of milk was made with different strengths, they were all used together to create this amazing masterpiece, the Mona Lisa.

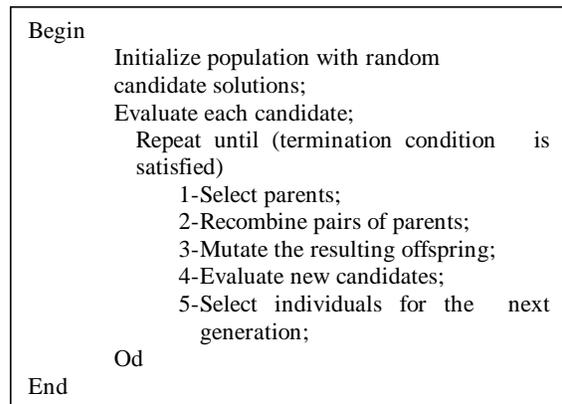
In this study, a specified target image will be generated, in a patch level format, from a given source image by applying the evolutionary algorithm (EA). Once the EA selects the best patch based on minimum error between the target and source patches, the luminance and chromatic values are transferred from source patch to the target patch.

This paper is structured as follows; section 2 presents the evolutionary algorithm EA concept, section 3 explains color space conversion and evolutionary algorithm operators in details, section 4 presents results of the proposed image generation algorithm, and finally the last section presents conclusions.

## 2. Evolutionary Algorithm (EA)

An EA is inspired by biological evaluation and is widely believed to be an effective global optimization algorithm. The inherent power of EAs lies in their ability to exploit accumulating information about an initially unknown domain in a highly efficient manner [4]. Therefore, EA was selected to design a search engine in this research. The search is implemented as an iterative procedure which can start with an arbitrary population which consists of a certain number of possible

individuals. If the search technique is properly designed and implemented, a satisfactory solution will be quickly found in the limited number of iterations. The individual with the best fitness is survived to the next generation by applying Evolutionary operators [8]. The general scheme of an EA can be given in Fig.(1) in a pseudo-code fashion [9].



**Fig.(1) : The general pseudo-code of an Evolutionary Algorithm scheme Algorithm in.**

At start, the mating selection is to distinguish among individuals based on their quality. An individual is a parent if it has been selected to undergo variation in order to create offspring. The parent selection is typically probabilistic. Thus, higher quality individuals get a higher chance to become parents than those with low quality. Next, recombination is an operator applied to two or more selected parents and results in one or more new children.

One variation of recombination operators is Uniform crossover, which works by treating each gene independently and making a random choice as to which parent it should be inherited from. This is implemented by generating a string of  $l$  random variables from a uniform distribution over  $[0, 1]$ , where  $l$  is the length of the chromosome.

Scanning crossover generalizes the standard uniform crossover. The simplest form of the scanning crossover mechanism works by taking  $n$  parent strings and creating one child through investigating the  $j$ th ( $j=1, \dots, l$ ) gene of the parents and choosing one of them to be the  $j$ th gene of the child. Notice that the

way the choice is made about the gene to be inherited is not specified. This allows different versions of gene scanning distinguished by different choice mechanisms. Possible problem independent choice mechanisms are for instance uniform random choice, voting (i.e. scanning for bit pattern representation, where the allele with the highest number of occurrences should be inserted in the child) or random choice biased by the fitness of the parents [10][11].

Finally, mutation is applied to one individual and results in one new individual. Executing recombination and mutation leads to a set of new offspring that compete based on their fitness with the old ones for a place in the next generation. This process can be iterated until an individual with sufficient quality (a solution) is found or a previously set computational limit is reached [9].

**3. The proposed Image Generation**

**Algorithm**

An EA applies the principles of evolution found in nature to solve the problem with the possibility to find optimal solution. The most popular type of EA is genetic algorithm which has a great deal of potentialities in scientific and engineering optimization. Moreover, GA is naturally applicable to process feature selection, as it has inherent parallelism and global optimization capability in an exponential search space [7].

The first step in the proposed algorithm is to convert both source and target images from the correlated RGB color space to a de-correlated space YIQ (luminance and chromatic values) as formulated in eq.1 [11].

$$\left. \begin{aligned} Y &= 0.30R + 0.59G + 0.11B \\ I &= -0.15R - 0.29G + 0.44B \\ Q &= 0.62R - 0.52G - 0.11B \end{aligned} \right\} \dots\dots\dots (1)$$

Next, one has to determine the patch size,  $w$  which can be any odd number such as  $3*3$ ,  $5*5$ , and  $7*7$  and so on. Both images are divided into many un-overlapped patches where the number of these patches is computed as in equ.2. One can consider  $w$  as the application control parameter. By this, we mean that the varying of  $w$  can greatly effect the quality of the final results. In the next

section, we will demonstrate the effect of this parameter.

$$\left. \begin{aligned} n_s &= \frac{width_s * height_s}{w * w} \\ n_t &= \frac{width_t * height_t}{w * w} \end{aligned} \right\} \dots\dots\dots (2)$$

Where:

- $n_s$  : Number of patches for source image,
- $n_t$  : Number of patches for target image,
- $width_s$  : Width of the source image,
- $width_t$  : Width of the target image,
- $height_s$  : Height of the source image,
- $height_t$  : Height of the target image, and
- $w$  : Patch width.

Then, proposed EA is started, the first step of EA is initializing the population with number of individuals (i.e. chromosomes). An individual is made up of a collection of genes which have been selected randomly and it's represent the parameters to be optimized.

gene (1,1)	gene (1,2)	---	gene (1,n)
gene (2,1)			⋮
⋮			
gene (m,1)	gene (m,2)		gene (m,n)

(a) An individual structure

pixel (xc-1,yc-1)	pixel (xc-1,yc)	pixel (xc-1,yc+1)
pixel (xc,yc-1)	pixel (xc,yc)	pixel (xc,yc+1)
pixel (xc+1,yc-1)	pixel (xc+1,yc)	pixel (xc+1,yc+1)

(b) Patch Structure

**Fig.(2): An individual structure with patch.**

$$\left. \begin{aligned} n &= \frac{width_t}{w} \\ m &= \frac{height_t}{w} \end{aligned} \right\} \dots\dots\dots (3)$$

$chrom\_size = m * n$

Each gene (i, j) value (i.e. allele) represents a center (xc, yc) for a specified source patch (see Fig.(2)). It depicts the structure of the individual where it is represented as two-dimension array in Fig.(2-a), while the array in Fig.(2-b) shows the patch structure. The collection of individuals made up the population with a specified size as in Fig.(3) so that each individual is converted into one-dimension array with gene indices (0 ...m\*n) and the last cell at each individual represent fitness computation (e.g., fit0) represents the fitness for individual0).

The second step of EA is to compute fitness function for each individual to evaluate the quality of individual. The fitness is computed by summation of luminance and standard deviation matching error between gene (i, j) which represent specific  $p_s(xc,yc)$  and  $p_t(i, j)$  as depicted in eq.5.

<i>Individual0</i>	gene(0,0)	gene(0,1)	---	gene (0, (chrom_size)-1)	Fit(0, chrom_size)
<i>Individual1</i>	gene(1,0)			⋮	⋮
	⋮				
<i>Individual49</i>	gene(49,0)	gene(49,1)	---	gene(49,( chrom_size)-1)	Fit(49, chrom_size)

**Fig.(3): The population representation with population size=50.**

$$gene - fitness(i, j) = 1 / (0.5(abs(L_{avs}(xc, yc) - L_{avt}(i, j))) + 0.5(abs(\delta_s(xc, yc) - \delta_t(i, j)))) \dots \dots \dots (5)$$

Where

$L_{avs}(xc, yc)$ : Luminance average for source patch referred to by value of gene (i, j).

$\delta_s(xc, yc)$ : Standard deviation for source patch referred to by value of gene (i, j).

$L_{avt}(i, j)$ : Luminance average for target patch being identify by gene (i, j).

$\delta_t(i, j)$ : Standard deviation for target patch being identify by gene (i, j).

This computation repeated for all genes in an individual to find the total error as depicted in eq.6.

$$individual - fitness = 1 / (\sum_{i=1}^{i=m} \sum_{j=1}^{j=n} gene - fitness(i, j)) \dots \dots \dots (6)$$

Third, EA operators are applied; there is no selection operator because we need all individuals to be the parents for the creation of the super offspring. These parents are recombined by multi sexual recombination to produce one super child as in Fig.(4). In this figure bestgene0 is chosen from the first column in Fig.(3) so that it is the closest gene to the  $p_{t0}$  and so on for the rest genes which are selected from corresponding columns in population.

The implemented recombination operator offers a tool for amplifying its effects and goes beyond the canonical perspective where the arity (i.e. the number of parents that it takes as inputs) equals to two parents (i.e. crossover). The resulting multi-parent recombination operator can be categorized by the mechanism for combining the information of the whole

population individuals. This mechanism generalized uniform scanning crossover based on best gene voting, so-called here as gene voting scanning recombination. The proposed gene voting scanning recombination works by taking the whole parent strings and creating one super child through investigating the *jth* ( $j=1, \dots, m \times n$ ) gene of the parents and choosing the best gene (with the best gene-fitness) to be the *jth* gene of this super child.

Next, mutation is the second variation operator, designed here to work on the generated super child and create from it a group of  $(pop_{size} - 1)$  new offspring by applying randomized change to the entire genotype representation of that super offspring. This mutation scheme works, as creep mutation, by adding a randomly small (positive or negative) value to each gene value. As a result one of the neighborhood patches (top, down, left, or right) of the specified gene's patch will be a candidate for draw. Then the general layout of the evolutionary algorithm for image generation:

After a specified number of generations is met, the genotype of best offspring is decoded to its corresponding phenotype. This is done by converting YIQ into RGB using eq.7.

$$\left. \begin{aligned} R &= Y + 0.956 + 0.62Q \\ G &= Y - 0.272I + 0.64Q \\ B &= Y - 1.108I + 1.705Q \end{aligned} \right\} \dots \dots \dots (7)$$

best gene (0)	best gene (1)	---	best gene (chrom_size-1)	Fit (chrom_size)
---------------	---------------	-----	--------------------------	------------------

**Fig.(4) : The offspring structure.**

The following algorithm describes the general layout of the evolutionary algorithm for image generation.

```

Algorithm: Evolutionary image generation.
Initialize: initiates the population with random generated chromosome;
p (0)={chrom(1)...chrom ( pop_size - 1)}
Evaluate fitness: evaluate chromosome fitness by evaluating each of its genes fitnesses.
  For k=1 to pop_size
  For i=1 to n
  For j=1 to m
  Evaluate fitness (chrom(k).gene(i,j))
  End:
  End
  Evaluate fitness (chrom(k))
  End
Repeat until (number generation > max_generation)
  Chrom(1)=recombine(chrom(1),...,chrom(pop_size))
  For i=2 to pop_size Do
  Chrom(i)=mutate(chrom(1))
  End
oD
    
```

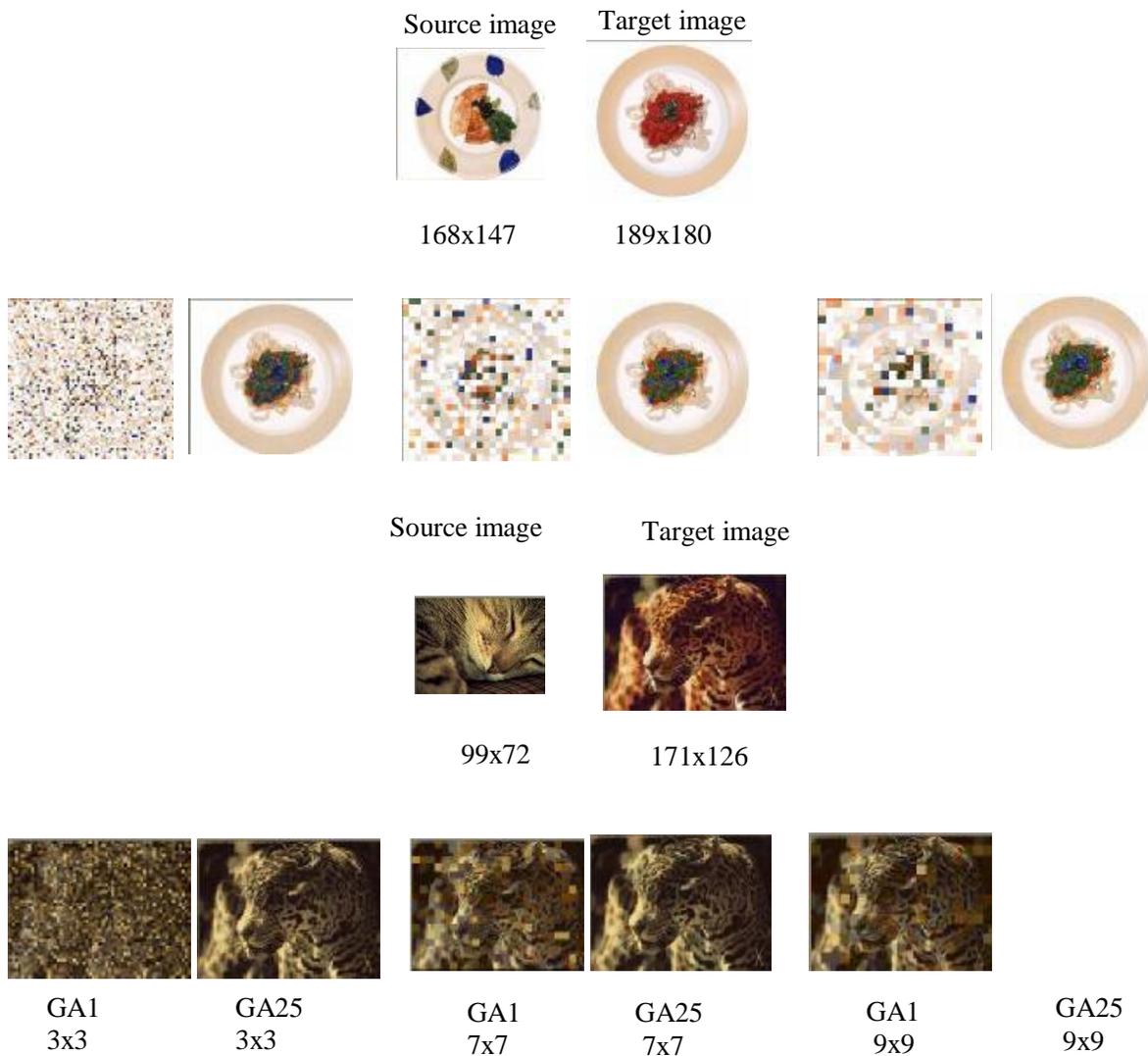
**4. Experimental Result**

This section presents some results on the usefulness of evolutionary based image generator when compared with the full search mechanism proposed by Ammar and aminna [5]. The proposed algorithm work with population size equals to 50, maximum number of generation is 25 and patch size is 3\*3, this setting is obtained from several EA running. There are many factors that can influence in the computation time such as, population size, number of generation, patch size and size of the target and source image. When the population size and the number of generation are increased, we can get better result but this process requires more time. Also, increasing the patch size can give result with more accuracy as shown in Table (1) (see Fig.(5)). Fig.(6) depicts the difference between (Fig.(6-a)) and (Fig.(6-b)) which are the best and worst image generated respectively. The best image is chosen depending on minimum error can be accessed at  $EA_{25}$  among all images as illustrated in Table (2) but the selection of worst image depending on maximum error. The multi sexual recombination operator is used in the proposed EA which can give more acceptable result than the uniform recombination (see Fig.(7)). Fig.(8) presents comparison results with exhaustive search method work by Ammar and

aminna [5]. Results of exhaustive search require more running time than the proposed EA with the same patch size. Table (2) presents the results of the EA at the first generation ( $EA_1$ ) and the final generation ( $EA_{25}$ ).

**Table (1)**  
*Show the MSR of the generated image when patch size are 3x3, 7x7 and 9x9.*

Source image	Target image	MSE		
		3x3	7x7	9x9
		4.5	0.9	0.8
		3.7	1.8	1.5

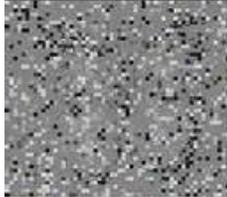
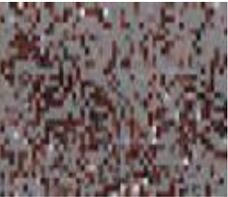
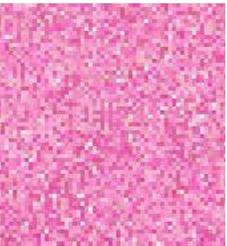
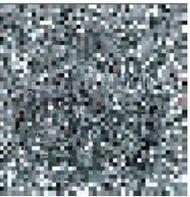


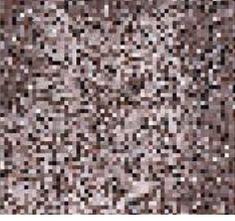
**Fig.(5):** Show the effect of varying patch size on the generated image.

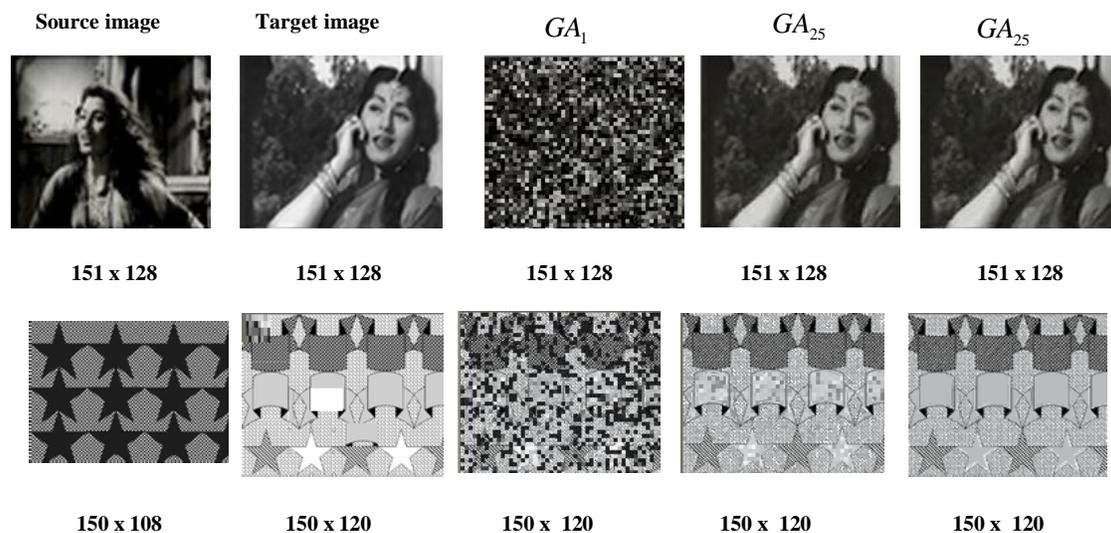
**Table (2)**  
**Comparison results between the EA and full search in computation time, and MSR.**

Source Image	Target Image	$EA_1$	$EA_{25}$	$MSE$ Full Search	$MSE$ EA	Time (sec.) Full Search	Time (sec.) EA
		6.51379	0.1228056	2.3	2.2	403	99
		4.60744	0.16211	1.8	1.9	999	127
		5.466416	0.1498	3.5	3.5	1302	153
		4.62681	0.04994	0.8	0.8	1634	186
		8.67139	0.379	7	6.8	827	99
		7.09317	0.118	2.6	2.5	1007	108
		7.50205	0.529	8.1	7.2	957	115
		6.675591	0.195	1.7	1.8	1085	165
		4.59196	0.06	1.3	1.6	537	99
		5.68735	0.239	2.97	7.5	455	101
		4.75831	0.158	3.138	2.3	798	107
		8.27968	3.174	6.92	17.6	643	98



Source image	Target image	$GA_1$	$GA_{25}$	Full search
 99 x 150	 120 x 150	 120 x 150	 120 x 150	 120 x 150
 168 x 150	 168 x 150	 168 x 150	 168 x 150	 168 x 150
 180 x 150	 168 x 150	 168 x 150	 168 x 150	 168 x 150
 165 x 180	 174 x 183	 174 x 183	 174 x 183	 174 x 183
 174 x 153	 138 x 141	 138 x 141	 138 x 141	 138 x 141

Source image	Target image	$GA_1$	$GA_{25}$	Full search
 159 x 180	 153 x 138	 153 x 138	 153 x 138	 153 x 138
 183 x 138	 153 x 151	 153 x 151	 153 x 151	 153 x 151
 144 x 168	 177 x 165	 177 x 165	 177 x 165	 177 x 165
 144 x 168	 177x165	 177 x 165	 177 x 165	 177 x 165
 147 x 117	 148 x 118	 148 x 118	 148 x 118	 148 x 118



**Fig.(8):** Some results of generated image obtained by the EA and full search algorithm. First column is the target image, 2nd column is the source image, 3rd column is generated image at the first population, 4<sup>th</sup> column is EA result and 5<sup>th</sup> column is full search result.

## 5. Conclusion

This paper presents how evolutionary algorithm can be used to generate a specified image (colored or gray) without any intervention from user. The results were satisfactory and achieved with less running time when compared with other image generation mechanism. Further the results show that there is some compromise between quantitative and qualitative results. Increasing patch size leads to decreasing the MSR but, on the other hand, better visual results can be achieved while decreasing patch size. Also the proposed EA can give good result by using multi-sexual recombination rather than uniform recombination.

## 6. References

- [1] V. Karthikeyani<sup>1</sup>, K. Duraiswamy<sup>2</sup> and P.Kamalakkannan<sup>3</sup>, 'Conversion of Gray-scale image to Color Image with and without Texture Synthesis', IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.4, April 2007.
- [2] Welsh, T., Ashikhmin, M. and Mueller, K. (2002) 'Transferring color to grey scale images', ACM Transactions on Graphics, Vol. 21, No. 3, July, pp.277–280.
- [3] Di Blasi, G. and Reforgiato, R.D. (2003) 'Fast colorization of gray images', Proceedings of Eurographics Italian Chapter, D.M.I., University of Catania, 2003.
- [4] Bara'a A. and Amina D.,(2007) 'Evolutionary colorization of grey scale images', Int. J. Advanced Media and Communication, Vol. 1, No. 3, pp.2164-2174, 2007.
- [5] Ammar A. and Amina D. 'Automated Approach For Color Image Generation', Journal of Engineering, Vol.14, No.1, 2008..
- [6] Mantas P.and Andrius U. 'A Survey of Genetic Algorithms Applications For Image Enhancement and Segmentation', Vilnius Gediminas Technical University, 2007, Vol.36, No.3.
- [7] Zhihua Xia,Xingming Sun,Jiaohua Qin and Changming Niu, 'Feature selection for Image Steganalysis using Hybrid Genetic Algorithm, Inform. Technol. J., Vol 8, No. 811-820, 2009.
- [8] Wen Chen, Yun Q. Shi, Guorong Xuan, Wei Su, "Computer Graphics Identification Using Genetic Algorithm", The 19th

- International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA.
- [9] A. E. Eiben and J. E. Smith, "Introduction to Evolutionary Computing", Springer 2003.
- [10] A. E. Eiben C. H. M. van Kemenade and J. N. Kok, A Orgy in the Computer Multi Parent Reproduction in Genetic Algorithms, European Conference on Artificial Life, Granada, 1995.
- [11] Gonzalez R.G., and Woods R.E. "Digital Image Processing", Wesley Publishing 2001.

### الخلاصة

يوضح هذا البحث خوارزمية سريعة لجمع طريقة النافذة الصغيرة والتطور معاً. تحاول هذه الخوارزمية معرفة النافذة الصغيرة الأكثر مناسبة عن طريق مقارنة الإضاءة بين صورة المصدر والهدف ويتم نقل معلومات الإضاءة و الألوان من صورة المصدر إلى صورة الهدف. يمكن لهذه الصورة أن تكون رمادية أو ملونة. في هذه الطريقة تظهر الصعوبات عند نقل أفضل إضاءة مع الألوان إلى صورة الهدف. يتم استخدام خوارزمية الوراثة للقضاء على هذه الصعوبات وإيجاد أفضل الحلول لهذه المشكلة عن طريق البحث عن أفضل نافذة بواسطة مطابقة الإضاءة إلى إن يتحقق عدد معين من الأجيال. فإذا كان الجيل الجديد يتألف من أولاد الجيل القديم فإن الإباء ستقتل وإذا لا فإن الإباء تبقى بالجيل الجديد وتنافس أبنائها. أثبتت النتائج بأن أفضل هدف ممكن إن نصل إليه وبأداء عالي وتحقق تقارب معقول عن طريق تقليص عدد المقارنات بين نقاط صورة الهدف و المصدر.