

## A FULLY AUTOMATIC GENETIC APPROACH FOR GRAYSCALE IMAGE COLORIZATION

Dr. Bara'a Ali Attea

Sana'a Khudayer Jaddwa Al-Janaby

University of Baghdad \ College of Science \ Computer Science

### ABSTRACT

Colorization is a computer assisted process of adding color to a monochrome (grayscale) image or movie. The early published methods to perform the image colorizing rely on heuristic techniques for choosing RGB colors from a global palette and applying them to regions of the target grayscale image. The main improvement of the proposed technique is the adoption in a fully automatic way the genetic algorithm as an efficient search method to find best match for each pixel in the target image. The proposed genetic algorithm evolves a population of randomly selected individuals (that represents a possible color setting for target image using a reference colored source image toward solution that could resemble natural or real colors to the objects of the target scene). Moreover this study proposes new crossover operator, called Spread out Uniform Crossover (SUX) that turns the recombination scheme of uniform crossover over spreading vital genes at the expense of lethal genes rather than exchanging genes between mating parents to the generated offspring. The results of the proposed colorization techniques are good and plausible.

### الخلاصة

التلوين هو عملية مساعدة الحاسبة بإضافة اللون إلى الصورة أو الأفلام الأحادية اللون (الرمادية). تعول الطرق المنشورة سابقاً لأداء عملية التلوين على التقنيات الحدسية لاختيار الألوان الأساسية (RGB colors) من لوحة ألوان الرسام وتطبيقها على مناطق الصورة الرمادية الهدف. التحسين الرئيسي للتقنية الجديدة هو اختيار الخوارزمية الجينية بشكل أوتوماتيكي كامل كطريقة بحث كفوءة لا يجاد التماثل الأحسن لكل عنصر صورة للصورة الهدف. تنشأ الطريقة الجينية المقترحة سكان لإفراد مختارون بشكل عشوائي (الذي يمثل تهيئة أو وضع اللون الممكن للصورة الهدف باستخدام مصدر - صورة ملونة - كمرجع نحو الحل الذي يمكن إن يشابه الألوان الطبيعية أو الحقيقية لكائنات المنظر الهدف). بالإضافة لذلك فإن هذه الدراسة تقترح عامل تهجين جديد يسمى تهجين النشر المنتظم (SUX) الذي يحول أسلوب التناسل للتهجين المنتظم عبر نشر الجينات الحيوية على حساب الجينات الميتة، بدل إن يبادل الجينات بين الأباء المتزوجون إلى الذرية المتولدة. أن نتائج تقنية التلوين المقترحة جيدة وقابلة للتصديق ظاهرياً.

## INTRODUCTION

Colorization is a term introduced by Wilson Markle in 1970 to describe the computer assisted process for adding color to black and white movies [Lev04]. His company Colorization, Inc. released the first full-length colorized movie in August 1985. Nowadays, the colorization term is used to describe the process of adding color to monochrome still images and movies [Sap04].

Colorization, in general, is an active and challenging area of research with a lot of interest in the image editing and compression community. With the luminance information and just some samples of the color ( much less than the ordinary sub-sampling in common compression schemes) , the color components of the data can be faithfully recovered .This has implications also in wireless image transmission, where lost image blocks can be recovered from the available channels [Sap04]. Additionally, colorization helps in scientific illustration by exploiting variations in chromaticity as well as luminance. Further, color can be added to a range of scientific images for illustrative and educational purposes. In medicine, image modalities which only acquire grayscale images such Magnetic Resonance Imaging (MRI), X-ray and Computerized Tomography (CT) images can be enhanced with color for presentations and demonstrations [Wel02].

Moreover, more "mundane" applications can benefit from colorization techniques. For instance, consider a scenario where two people that chat regularly through the Internet decide to enhance their virtual meetings with live video. If colorization software, inexpensive and fully automatic one, was available to them, they might buy less expensive monochromatic web-cams instead of color ones, use limited bandwidth by transmitting monochromatic video, but still be able to view fully colored video streams [Vie03].

## PREVIOUS WORK

Work interest in grayscale image colorization problem are pointed out:

- ❖ E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley [Rei01], describe a method for a general color correction that borrows one image's color characteristics from another using mean and standard deviation statistical analysis with  $\lambda\beta$  color space to make a synthetic image take on another image's look and feel. The transfer of statistics can fail in case that source and target images don't work well together. To remedy this, the user selects separate swatches, the algorithm then computes their statistics. Then, scales and shifts each pixel in the input image according to the statistics associated with each of the cluster pairs.
- ❖ T. Welsh, M. Ashikhmin, and K. Muller [Wel02], introduce a general semi-automatic t color transfer between a source color image and a destination grayscale image. They choose to transfer only chromatic information and retain the original luminance values of the target image. Further, their procedure is enhanced by allowing the user to match areas of the two images with rectangular swatches. Colors are then transferred but between the corresponding swatches.
- ❖ G.Di. Blasi, and R. D. Recupero [Bla03], propose a semi-automatic colorization method. In this approach, image pixels are grouped into clusters of bounded radius by the Antipolee Tree Clustering. This clustering algorithm works in such a way that "far" elements lie in different clusters. The RGB source image is first converted to YUV space and the Antipole tree is constructed, in which each vector contains the information necessary to perform the Antipole search and the UV components of the pixel color. Then, for each grayscale pixel, in scan-line order, construct its vector and perform the Antipole search to select the best matching vector in the Antipole tree and transfer its UV components to the Y component (luminance) of that gray pixel.
- ❖ L.F.M. Vieira, R. D. Vilela, and E. R. do Nascimento [Vie03], introduce a methodology for adding color to grayscale images from a database of color images. Initially the database is semantically indexed. The color transfer process then takes the grayscale image and the color one that has the most similar feature vector as inputs, and adds to each scalar pixel of the

former, the chromatic components of an automatically chosen pixel of the later. The colorization proceeds by scanning the target image in scan-line order so that, for everyone of its pixels, the source image pixel that is closest in terms of intensity value and intensity standard deviation is selected as a match.

- ❖ Levin, D. Lischinski, and Y. Weiss [Lev04], present a method based on the simple premise: neighboring pixels in space-time that have similar intensities should have similar colors. This premise is formalized by using a quadratic cost function and obtain an optimization problem that can be solved efficiently using standard techniques.
- ❖ Z. Pan, Z. Dong, and M. Zhang [Pan04], present a method to transfer color from a reference image to the whole video. This method take advantage of the correlation between two conjoint frames of video, by tracking the object and assigning the color of it in the preceding frame to that of the posterior one. In many cases, the background will keep unchanged while an object is moving. Rather than choosing RGB colors from a palette to color individual components, they choose  $\ell\alpha\beta$  color space that minimize correlation between channels for many natural scenes.
- ❖ D. Šýkorá, J. Buriánek, and J. Žára [Šyk04], present a novel color-by example technique which combines image segmentation, patch-based sampling and probabilistic reasoning .This method is able to automatic colorization of black and –white cartoon. They state the problem as: having two segmented frames. First frame serves as a color example where each region has one color index assigned from the user-defined palette. The second frame contains unlabelled target regions. The colorization task, then, is to assign color indices to target regions similarly to as they are assigned in the example frame.
- ❖ G.Sapiro [Sap04], presented an approach for adding color to a monochrome image or movie, based on considering the geometry and structure of the monochrome luminance input, given by its gradient information. The luminance channel in  $YC_bC_r$  color space faithfully represents the geometry of the whole (vectorial) color image. In addition to having the monochrome (luminance) channel, the user provides a few strokes of color, that need to be propagated to the whole color channels. The color is then obtained by solving a partial differential equation that propagates the user selected colors while considering the gradient information brought in by the monochrome data.

### THE COLORIZATION PROBLEM COMPLEXITY

The problem of coloring a grayscale image involves assigning three-dimensional (RGB) pixel values to an image whose elements (pixels) are characterized only by one feature (luminance). Since different colors may carry the same luminance in spite of differences in hue and/or saturation, the problem of colorizing gray-scaled images has no inherently "correct" solution [Bla03]. Thus , this 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 [Vie03]. Even in the case of pseudo coloring, where the mapping of luminance values to color values is automatic, the choice of the color map is commonly determined by human decision [Bla03].

A major difficulty with colorization lies in the fact that it is an expensive and time-consuming process. For example, classically colorization is done by first segmenting the image into regions, and then proceeds to assign colors to each segment [Sap04]. Unfortunately, automatic segmentation algorithms often fail to correctly identify fuzzy or complex region boundaries [Lev04]. Colorization of movies requires, in addition, tracking regions across the frames of a shot. Existing tracking algorithms typically fail to robustly track non-rigid regions, again requiring massive user intervention in the process [Lev04].

## THE PROPOSED COLORIZATION SYSTEM

The general procedure for coloring grayscale image requires as input a pair of images. The source colored image,  $S$ , and the target grayscale image,  $T$ . The source image could be selected to have similar semantic features to that of the target image (e.g., one can choose pair of faces, sunsets, trees, cars, etc.). Next, both source (colored) and target (grayscale) RGB images are converted to a de-correlated color space. The color space provides three de-correlated, principal channels corresponding to an achromatic luminance  $L$  channel, and two chromatic channels  $C1$  and  $C2$  in which changes made in one color channel should minimally affect values in the other channels. Examples of de-correlated color spaces are  $YIQ$ ,  $YUV$ ,  $\ell \alpha \beta$ , and  $CIECAM97s$ . The remaining (main) steps of the proposed colorization system are presented in what follows.

## INDIVIDUAL REPRESENTATION AND INITIALIZATION

First, a population of  $P_{size}$  random chromosomes is generated. Each chromosome in the population is coded as a two-dimensional  $m_T * n_T$  formation where  $m_T$  and  $n_T$  are respectively height and width of the target grayscale image. Each gene in the chromosome can hold an integer value,  $I$ , refers to a colored pixel in the source image. Hence,  $1 \leq I \leq m_S * n_S$  where  $m_S$  and  $n_S$  are respectively height and width of the source colored image. **Fig. (1)** depicts this chromosome representation:

	1	2	3	...	$n_T$
1	20	315	1	...	511
2	5	16	900	...	100
3	512	322	262	...	104
.	.	.	.	...	.
.	.	.	.	...	.
.	.	.	.	...	.
$m_T$	70	83	69	...	2

$m_T * n_T$

Fig. (1) The chromosome of height  $m_T$  and width  $n_T$  equal to the size of the grayscale image. Gene no. (1,1) in the figure, e.g., hold an integer 20 which refer to the colored pixel number 20 (from left-to-right and top-to bottom counting) in the source image .

## FITNESS EVALUATION

A GA must qualify its population individuals through fitness measure to let evolution to those favorite individuals at the expense of weaker ones.

A similarity metric can be calculated for each GA individual to determine luminance differences between the GA selected pixels and the target pixels. Objective function, here, is calculated in three different variations. In the first approach (named objective 1), luminance values difference is used as the similarity measure between target and source images in which small difference gets better similarity measure than large differences. As a whole, objective1 is calculated as the sum of (achromatic) luminance differences between target gray pixels (indicated by the gene numbers) and source colored pixels (indicated by the gene contents) :

$$\text{Objective 1} = \sum_{i=1}^{n_T} \sum_{j=1}^{m_T} |L_T(i, j) - L_S(I(i, j))| \quad (1)$$

where:

$n_T$ , and  $m_T$  are respectively width and height of  $T$  image .

$L_T$  ,  $L_S$  are respectively luminance values of  $T$  and  $S$  images.

$i, j$  are indices of a gene in row  $i$  and column  $j$  .

$I(i, j)$  an indirect index to a pixel in  $S$  .

In the second and third Objective calculation approaches, the texture similarity or match is also compared. Objective2 sums differences between luminance average of surrounding neighborhood  $N_T$  in  $T$  image and surrounding neighborhood  $N_S$  in  $S$  image:

$$\text{Objective2} = \sum_{i=1}^{n_T} \sum_{j=1}^{m_T} |L_T(N_T(i, j)) - L_S(N_S(I(i, j)))| \quad (2)$$

where :

$n_T$ ,  $m_T$ ,  $i$ ,  $j$  and  $I(i, j)$  are as defined previously in (1) .

$N$  is the surrounding  $3*3$  ,  $5*5$  , or  $7*7$  neighborhood .

$L_T$  is the luminance average of  $N_T$  neighborhood gray pixels .

$L_S$  is the luminance average of  $N_S$  neighborhood colored pixels .

Finally , Objective3 uses matches luminance differences in term of weighted average (50% weight) and weighted standard deviation(50% weight) of neighborhood pixels :

$$\text{Objective 3} = \sum_{i=1}^{n_T} \sum_{j=1}^{m_T} |L_T(N_T(i, j)) - L_S(N_S(I(i, j)))| \quad (3)$$

where :

$n_T$  and  $m_T$ ,  $i$ ,  $j$ ,  $I(i, j)$ , and  $N$  are as defined previously in (1) and (2) .

$L_T$  is the sum of 50% luminance average and 50% standard deviation of  $N_T$  neighborhood gray pixels .

$L_S$  is the sum of 50% luminance average and 50% standard deviation of  $N_S$  neighborhood colored pixels .

In Objective2 and Objective3, different colored regions give rise to distinct luminance clusters, or possess distinct regions. In other words, we expect pixels which are similar in texture to the source colored image regions to be colorized similarly. As high fitness values can represent good solutions to the problem at hand , the fitness then can be computed as inverse of objective function, i.e. ,

$$\text{Fitness} = \frac{1}{\text{Objective}} \quad (4)$$

## SELECTION OPERATOR

Once fitness is evaluated for each individual, we can form the mating pool that is a collection of individuals who will have the right to reproduce themselves into future generations. This selection is random, not deterministic. The most common selection operator used in GA is tournament selection. In tournament selection with size two, pair of chromosomes is compared. The string with highest fitness is copied into the mating pool. Repeating this process until the mating pool is filled with probably better chromosomes of  $p_{size}$  parents.

### CROSSOVER OPERATOR

After the mating pool is formed, pairs of chromosomes (parents) are randomly chosen to be manipulated by the crossover operator before releasing them into the offspring. There are various crossover operators (e.g., one-point, two-point, and uniform crossovers), but all aim at recombining the genetic material of the two selected parents in an efficient manner.

However, a modification to the uniform crossover is proposed here. The new crossover operator (as we called Spread-out Uniform Crossover **SUX**) modifies the uniform crossover to spread out vital genes at the expense of lethal genes rather than exchanging genes between mating parents.

A gene can be classified as vital or lethal according to a predetermined threshold value. The threshold value determines whether luminance difference between a gray pixel and the GA selected colored pixel is accepted or not. If difference is less than or equal threshold value, then we can regard this gene as vital and a copy of it can be propagated to both offspring, otherwise the gene is lethal and is insufficient to be propagated next. Fig. (3).6 below depicts the scheme of **SUX** operator.

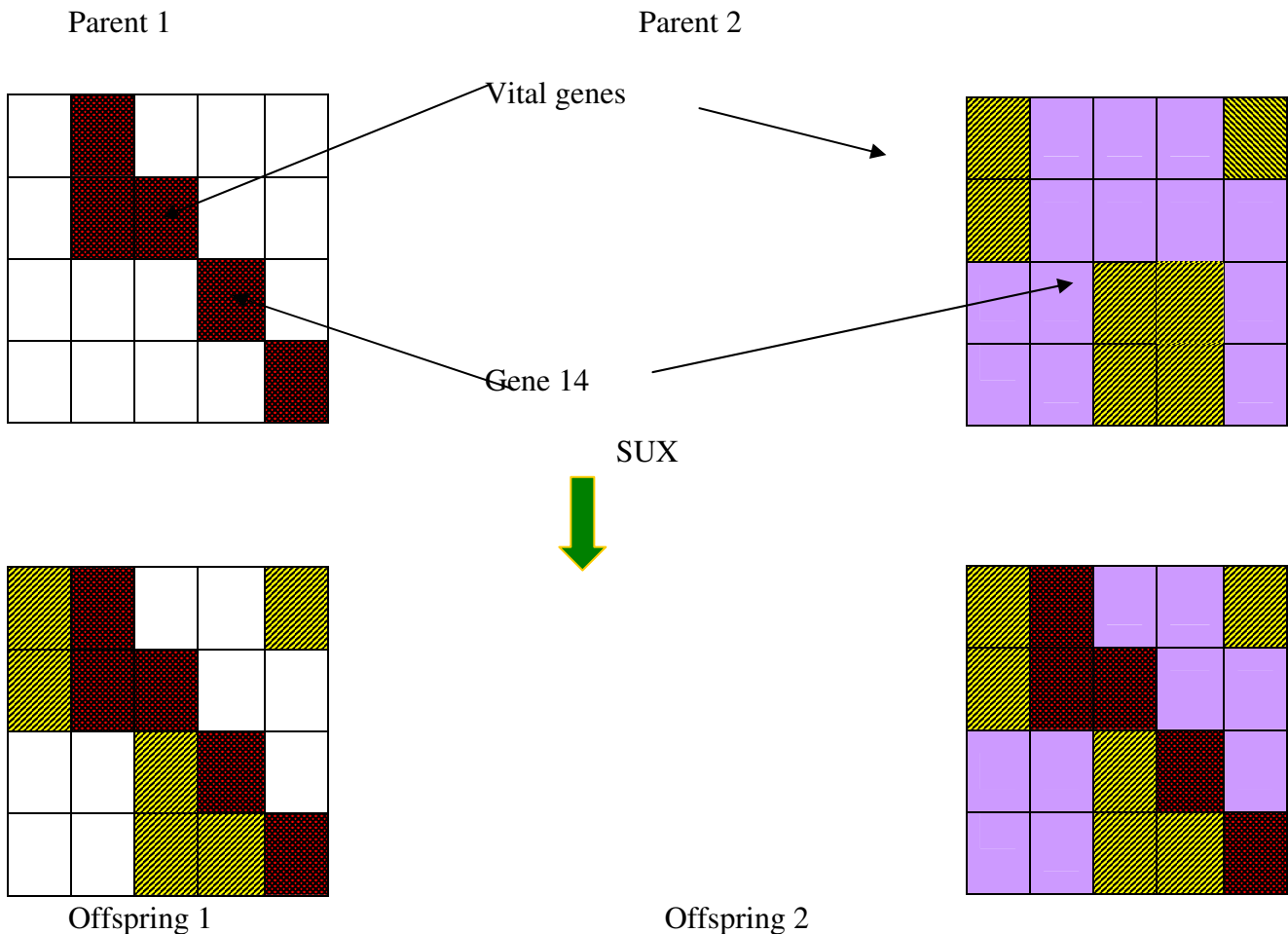


Fig. (2) SUX operator.

As can be seen from the figure, the vital genes of parent1 and parent2 are spread out over the generated offsprings. Moreover, SUX operator allows competition between two vital genes of one location to propagate the winner. However, lethal genes remain in their chromosome to its offspring without propagating to the second offspring. Additionally, for crossover operator there is a very important parameter that is crossover probability  $P_c$ . The normal range for  $P_c$  is between 0.5 to 0.8.

### MUTATION OPERATOR

The second perturbation operator that manipulates the genetic material of individuals is mutation, which traditionally in GA plays as background operator. It randomly alters the genes of a population with a predetermined (normally low) probability,  $p_m$  from 0.005 to 0.05 to avoid the algorithm from becoming a purely random search. A gene in the chromosome may be altered to any value (from 1 to the  $m_s * n_s$ , full size of source colored image) if the condition of mutation is satisfied.

### ELITISM STRATEGY AND STOPPING CONDITION

Elitist selection is the idea of preserving the best individual of a population in future generations. Elitist GAs copy the individual with the best fitness encountered so far into the next generation.

Different stopping conditions are used in GA literatures. However, we use the most common one. The GA is allowed to run to a maximum number gen-max, of generations [Gol89].

### CONVERGENCE OF THE PROPOSED COLORIZATION SYSTEM

The genetic algorithm introduced in the previous section employs exhaustive searching (or explore the luminance search space of the source colored image), which makes it slow. To compensate the lack of human intervention and at the same time accelerate GA speed, an acceleration approach is proposed here. Acceleration works by considering neighborhoods  $N_s$  source colored pixels with their luminance values given the luminance setting selected by the genetic algorithm for a given individual. The proposed acceleration can be formulated as:

1. Determine the size of the neighborhood  $N_s$ , e.g.,  $3 * 3$ ,  $5 * 5$ , etc.
2. Select the best GA individual to be modified.

For each gene content of the selected GA individual, the acceleration searches over those  $N_s$  neighborhood of the colored pixel indexed by the gene content,  $I$ , and replace it with the index,  $J$ , of the neighborhood pixel that has luminance value close to the luminance value of the gray pixel indicated by that gene.

By this way, the search technique imposed by the proposed colorization system has two forms. A global search ability manifested by the genetic algorithm to explore the luminance search space of the source colored image to locate suitable regions (or swatches) to that of the luminance regions of the target grayscale image. This global search is hybrid with a local search that exploit the selected GA swatches for most suitable pixels luminances to that of the corresponding pixels in the target grayscale image. In short, there is a collaboration between the explorative power of the GA and exploitative power of the acceleration search.

### RECOVER COLOR INFORMATION

The last step to do is that how to display the target image on the screen as a colored RGB one. This step represent how to extract phenotype of a GA individual and displaying it in RGB space on the screen. Figure 3.9 illustrates the color recovering sequences.

Recall that the target and the source images are represented in RGB color model. After processing (by GA and local accelerator) in luminance space, we can recover the color at each gray pixel by copying the two chromatic channels ( $C1_s$  and  $C2_s$ ) of the genetically chosen selected source colored image pixels into the output target image, combining them with the target luminance channel  $L_T$ , and then followed by a conversion back to RGB color space.

### RESULTS

The experiments have been performed on a database of different image classes with a total of 30 images of colored and grayscale natural scenes. Both source colored and target grayscale images of a colorization query are taken from this database. In general, database images can be divided into

two groups: homogeneous and heterogeneous. Images that include a single object in the foreground and this object is clearly discernible from a mostly homogeneous background is said to be homogeneous (e.g. Apple, sunset1, and tiger1). On the other hand, an image that has multiple objects on the foreground, or has a cluttered background, or is illuminated in an uneven way is said to be a heterogeneous image (e.g., Trees, White Rose, and yellow Roses). Note that some images may not conform to this loose definition of the homogeneous or heterogeneous group. In other words, they could be semi-homogeneous or semi-heterogeneous images. Intuitively, for the colorization problem, a heterogeneous image is more complex than a homogeneous one. A colorization query requires two images, one source colored image, and one target grayscale image. Although input images can be taken from different classes, and one can just to see how this colorization technique work, we select input source and target images to be of the same semantic meaning. For example, the input to the colorization query can be a pair of cats, a pair of sunsets, etc.

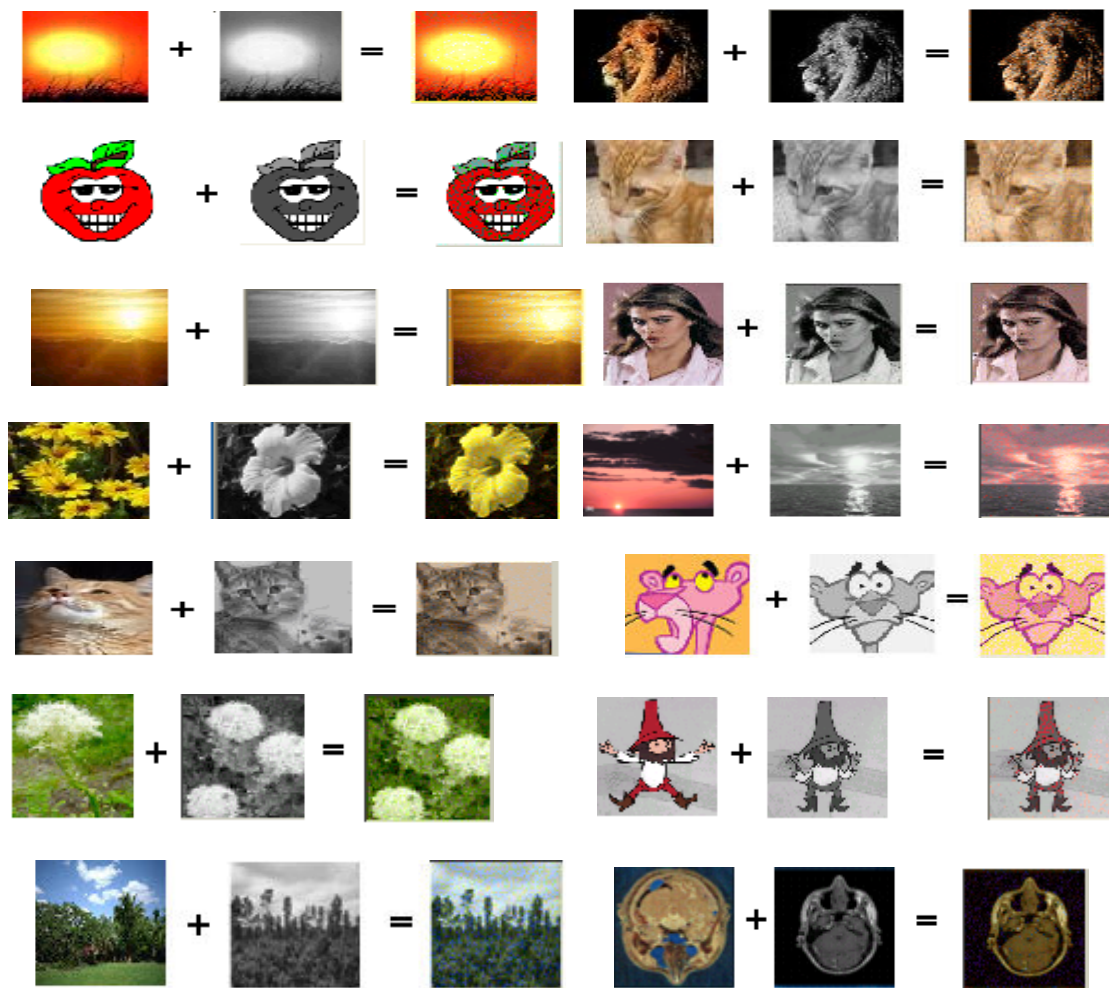


Fig. (3) Results (source colored image + target grayscale image = target colored image).

## CONCLUSIONS

The presented results analyze the behavior of the genetic algorithm for grayscale image colorization problem experimentally. The core of the proposed colorization technique consists of searching luminance in a high dimensional space using genetic algorithm. The genetic algorithm searches for each target pixel, the most suitable source pixel that matches in luminance value. Take a simple example the case in which the source and target images are both of size  $32 \times 32$  pixels. Then for each target pixel of the 1024 ( $32 \times 32$ ) pixels, the genetic algorithm explores the 1024 candidate





source pixels for best possible luminance match. This leads to a huge search space size of  $1024^{1024}$  potential solutions. Taking this problem complexity into account, it can be argued that the proposed colorization technique achieves convergence to suitable and near realistic results in a fully automatic way (without human intervention) and without adding excessive iterations.

The results of the proposed colorization system, as a whole, demonstrate that it is possible to color many grayscale images (including natural scenes, cartoons, and scientific data) in a way that is completely automatic and look, reasonably well using a GA of small population size and a few generations evolution. Further, this technique works well on scenes where the image is divided into distinct luminance clusters. More images can be colorized using this technique with better results but at the expense of increasing the GA population size and/or generations number. The satisfactory quality of the technique is established by collaborating the exploration power of the genetic algorithm with the exploitation power of the local acceleration search operator. One can imagine that the GA can locate the most suitable clusters (in luminance and / or texture) to those of target image while the local accelerator searches the located GA clusters for most suitable pixel (in luminance) to transfer its color to the corresponding grayscale pixels.

## REFERENCES

- DiB, G., Reforgiato, R.D., (2003), Fast Colorization of Gray Images, Eurographics, Italian Chapter.
- Goldberg, D. L., (1989), Genetic Algorithms in Search Optimization, and Machine learning,, Addison-Wesley, USA.
- Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H., (2001), Image Analogies, In Proceeding of ACM SIGGRAPH ,pp. 327-340.
- Levin, A., Lischinski, D. and Weiss, Y., (2004), Colorization using Optimization, ACM Transactions on Graphics.
- Pan, Z. Dong, Z. and Zhang, M., (2004), A new Algorithm for Adding Color to Video or Animation Clips, In Proceedings of WSCG-International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 515-519.
- Reinhard, E., (2001), Statistical approaches to image and scene manipulation, ACM SIGGRAPH/EUROGRAPHICS Campfire, Snowbird Utah.
- Sapiro, G., (2004), Inpainting The Colors, IMA Preprint Series #1979 ,May.
- Sýkora, D., Buriānek, J. and Žāra, J., (2004), Unsupervised Colorization of Black-and-White Cartoons, Proceedings of NPAR, Annecy, ACM SIGGRAPH .
- Vieira, L.F.M, Vilela, R.D., Nascimento, E.R., (2003), Automatically choosing source color images for coloring grayscale images.
- Welsh, T., Ashikhmin, M., and Mueller, K., (2002), Transferring Color to Greyscale Images, In Proceeding of ACM SIGGRAPH , pp.277-280.