FITNESS FUNCTION FOR RENDERING IN THE LOOP DIFFERENTIAL EVOLUTION ALGORITHM

[1]M. OLGUIN-CARBAJAL, [2]R. BARRON-FERNANDEZ, [2]J. L. OROPEZA-RODRIGEZ, [1]J. ALVAREZ-CEDILLO AND [1]J. C. HERRERA-LOZADA

[1]Center for Innovation and Development in Computer Technology, National Polytechnic Institute
[2]Center of Investigation in Computation, National Polytechnic Institute
E-mail: {molguin, jlozada; jaalvarez; }@ipn.mx
{rbarron, joropeza;}@cic.ipn.mx

ABSTRACT

Optical properties of scenario objects for rendering realistic images synthesis are highly important for computer graphics. We have proposed a methodology to obtain some illumination parameters of a real scenario represented by an acquired image and use these parameters for the adequate rendering of an equivalent virtual scenario made by hand. The proposed methodology uses an acquired image that we call ObjetiveImage and compare it with a rendered equivalent image of a virtual scenario, called RenderedImage that represents the original. The real scenario is approached by hand, but some of the illumination parameters like light source position and intensity as well as objects color are searched by using a Differential Evolution algorithm. Rendering images that approximate to the real acquired image, by some virtual scenario parameter modification based on the DE search, we call "Rendering in the loop". Finally we use the obtained parameters to render a better resolution image to obtain the result. The fitness function for the comparison of the two images is the most important search in this investigation.

Keywords: *Differential evolution, rendering, loop, optical.*

1. INTRODUCTION

Computer graphics rendered images have been evolving since 1970 when the first CGI were presented on an oscilloscope screen. Today the CGI are using rendering algorithms with illumination models (IM) that generate really impressive images. To obtain the most realistic images it is necessary to use these IM by using optical parameters of objects closer to the real ones, like reflection index, refraction index, absorption index to red, green and blue, etc. Finding good values for these parameters gives better rendered images. There exist mainly three methods to find the optical values of objects: one is by direct measurement, two by mathematical approximation, and three by estimation.

The methodology that we propose is an optical parameter of objects estimation by using differential evolution (DE) individuals as optimization method, fig. 1. By comparing a rendered image with test values proposed by each individual, versus an acquired image obtained with a single camera. Other estimation methods use an open loop which means it has no feedback. On the other hand, DE, by its own nature, is an optimization method with feedback. Therefore we put a rendering algorithm as part of the fitness function so we have a rendering in the loop estimation method.

There already exists some methods that use evolutionary techniques for optical parameters estimation, but these proposals use a human decision as part of its fitness function [1].

The main hypothesis of this article is to prove that this methodology works by using a very simple scenario with a simple illumination model. The objective is to find the X, Y, and Z position as intensity for the light source and the color of the objects in the scenario, the sphere and the plane.

2. RELATED WORK

Developed investigations to obtain optical properties try to measure object color, refraction index, reflection index, textures acquisition, media scattering properties, and so on, correctly. For example Furukawa et al. [2] uses an acquisition system based on a wide range laser to acquire a 3D model object texture. Also using a parameter acquisition system Gero Muller et al. [3] had a hemisphere with camera and fixed light sources. This system uses massive parallel processing and has no moving parts. It obtains a 3D texture as well as object color and reflection index.

Other approaches use 6 high resolution digital cameras and a light array, used by Wojciech Matusik et al. [4]. They put objects on a spinning table and by using a multi-background for alpha matte acquisition, and a matte environment for multiple viewpoints. To achieve a tridi-mensional reconstruction of appearance, color, refraction and reflection index. G. Muller et al. [5] designed a system with a single camera and a light source at a fixed position while the camera moves in a semicircle to obtain color parameters, as well as 2D texture reflection index of realistic materials with varying conditions. This technique gives very good results. However, it uses a lot of processing power and is time consuming. Also the great amount of generated data makes acquired data compression necessary, and according to Wai Kit Add Ngan [6] the uncompressed data needs about a Gigabyte storing space for 81 views for a texture patch of 256X256.

A very simple but powerful technique for optical parameters estimation has been developed, Srinivasa G. Narasimhan et al. [7] uses scattering properties estimation on different diluted media. They use a very simple device and technique once parameters had been estimated. This can be used in a Montecarlo renderer for a photorealistic image generation.

All previous techniques use an acquisition of optical parameters and then use it for image rendering. Our proposal is very different. By using a loop process to estimate illumination parameters for rendering we acquire only a single image and use it as reference for a rendering approximation using bio-inspired heuristic.

A. Evolutionary rendering

Evolutionary algorithms in image rendering had been used before in many aspects. In 2D painterly rendering, John Collomose says ``apply Genetic Algorithms for style selection using interactive aesthetic evaluation" [1], and allow the user to select visual styles like expressionism, pointillism, impressionism, and abstract. They use low-level parameter selection to tune the visual style. Also Collomose and Hall [8] had developed an automatic system for adaptive painting applying evolutionary search techniques to machine learning, by using Genetic Algorithms as a mechanism to automatically level control of the detail in paintings, but only in a single painting style. Tatsuo Unemi is an evolutionary artist who uses mathematical expressions to generate images by using different versions of his SBART software [9] and [10].

In 3D artistic rendering scientists and artists use a wide range of techniques to evolve 3D geometry in various domains. William Latham and Stephen Todd worked on the earliest organic forms created using their software in 1990 [11]. Later, Rowbottom's Form software made a PC-based implementation of Latham's approach [12]. Latham and Todd expanded and made a PC Mutator system to allow their interactive genetic approach to interface to other PC packages like drawing tools [13].

For the rendering process, evolutionary algorithms have been used to obtain a better beam distribution in a raytracing renderer by the super-sampling of a light beam, like that used by Dutre, Suykens, and Willems [14]. Beyer and Lange have applied genetic algorithms to solve the problem of integrating radiance over a hemisphere [15]. They used the individuals as rays, which evolved to improve their distribution on a surface point with respect to the incident radiance. Another approach is that used by Veach and Guibas, where the beam paths are selected using a Metropolis method to mutate the paths instead of the most commonly Montecarlo used technique. Their algorithm starts from a few light transport paths and applies random mutations. As result ``in the steady state, the resulting Markov chain visits each path with a probability proportional to that path's contribution to the image" [16].

Our proposal is different because we use the evolutionary and the rendering algorithm, as a tool to estimate the optical parameters for a better approximation to a real acquired image. In our case neither for path beam distribution nor for generating artistic images.

3. RENDERING IN THE LOOP

Our proposed methodology consists of an optical parameter optimization loop that we call "Rendering in the loop", which consists of four steps:

- Acquire a single reference image containing parameters of objects.
- Build by hand an equivalent scenario.
- Use a bio-inspired optimization function inside a renderer to estimate parameters.
- Use obtained optical parameters for complex object and scenarios rendering.

This way the proposed methodology can obtain the searched parameters, fig. 1. To test the methodology we programmed a simple ray tracer algorithm to render the test image. One we call RenderedImage. We use a very simple illumination model considering: one single light source, objects diffuse color, specular reflection and a constant ambient light. Then we built a simple acquisition device with matte background to obtain the reference image that we call ObjetiveImage.

A. Development

We developed two comparison functions between two different pixels from two images; one is based on the absolute value of the subtraction of ObjetiveImage and RenderedImage, eq. 1. Also we included a differential evolution algorithm as multi-objective optimization bio-inspired heuristic. This function obtains good values for the light source, like position and intensity, but did not find good color values for the objects.

$$Dif = \sum_{i=0}^{i=n} |RGBObj_i - RBGTray_i|$$

Where:

- n = Xmax * Y max
- RGBObji = Each pixel i component objective image value
- RGBTrazai = Each pixel i component rendered image value

In the second comparison function, every RGB component is treated as a single objective, and we use ten pixels for each image. Comparison function gives one value for each RGB component. With a high component value, the difference between both images is large, and with a little component value the difference is small.

Every individual vector has all the needed optical parameter set for the illumination model. By using it, the rendering algorithm generates a set of two pixels, each one representing a part of the image. To obtain individual aptitude we compare a ray traced rendered set of pixels with their corresponding ones in the acquired image. As the differences decrease in every RGB component, the individual fitness increases. The two pixel selection is to obtain an image interest area denoted for two independent image elements. Every vector has the color components for the sphere. This comparison function find good values for the color of the objects, however, did not find good values for the light source. So, we use these two comparison functions one after the other, first one find the light source values (X, Y, and position, as well as intensity). Then, the second one, using the light parameters found out by the previous step, find good value for the color of the objects, fig 2.

B. Algorithm

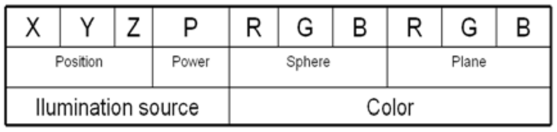
The proposed algorithm receives a reference image, obtained by the acquisition system and stored into a 128x96 matrix called MatrixObj. Then we initialize the DE (Differential evolution) individuals in order to generate population to evaluate the aptitude of each individual. The evaluation consists of two steps.

- By using the X vector of one chosen individual, we assign to each variable vector a parameter from a virtual world. The virtual world is used as an input for a ray traced algorithm to generate an image of 128 x 96, and the image is stored into a matrix named MatrizTray.
- Then we call comparison function for two images, taking as input both matrixes MatrixObj and MatrixTray to generate three numeric values, which is the difference between two image RGB values. These values will be stored as a particle aptitude.

The DE evaluates every individual and selects the best. The best individual information will be used by DE to save results in a file of best per generation individuals. At the same time we store in a BMP file the best individual's values from each generation. Only one image will be saved per each 5 generations.

The variables vector of each DE individual has a correspondence with the virtual world light source, so the first three variables will have the X, Y and Z values of the light source (pos L). The fourth variable will be the intensity source (Power L). Other variables are the RGB sphere values (sph) and the floor (pla), so the X vector of each DE individual will correspond to:

[Xpos_L, Ypos_L, Zpos_L, Power_L, R_sph, G_sph, B_sph, R_pla, G_pla, B_pla]



X	Y	Z	P	R	G	B	R	G	B
Position			Power	Sphere			Plane		
Ilumination source				Color					

FIGURE 3.Vector containing scenario optical parameters for each individual.

The implemented DE is a multi-objective algorithm looking to minimize three RGB values, each one corresponding to a function to minimize and the generated image from the ray tracing. We have a light dominance, considering two or three elements. We have conducted ten program runs, with a stop criterion of 200 generations, with 60 individuals. By using a CR value of 0.8 and F of 0.6, we present the results of one run, as follows (fig 4).

4. EXPERIMENTS AND RESULTS

We made a set of tests with red, green, blue, yellow, and metallic separated spheres. In this particular case, the optical parameters to estimate were: first light source position as well as intensity. Second, objects color. We showed results by using a yellow sphere as ObjetiveImage, Fig. 4.

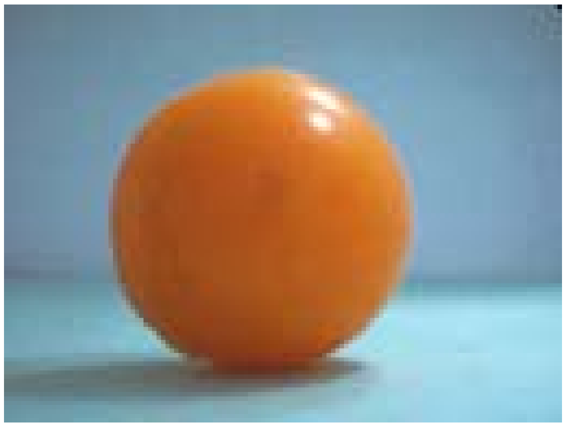


FIGURE 4. Acquired image, ObjetiveImage.

As we can see, the first step uses the comparison function that obtains the light source position and intensity. We can see that the shadow is near to that of ObjetiveImage. After PAS00201, the images appear with color, because from here we use the second comparison function to estimate the colors for the sphere and for the plane. At firs the rendered image seems with wrong colors. Slowly each image is better than the previous. Then we can see that all images have good light intensity, color, and the shadow is in the right place, fig 5.

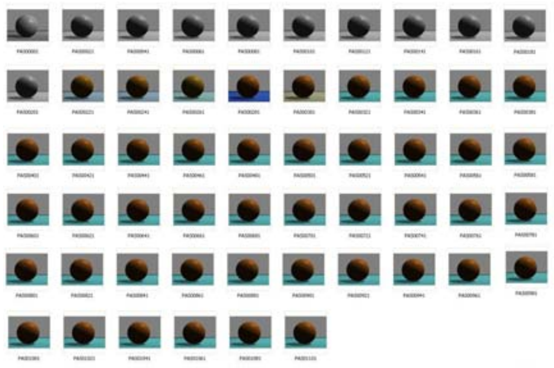


FIGURE 5. Different rendered images obtained by parameter evolution.

We can see the final rendered image near to the ObjetiveImage and see the result, fig. 6. The color is not the same because the illumination model we use is a simple one, but the main color for the floor and for the sphere are in the rigth way.

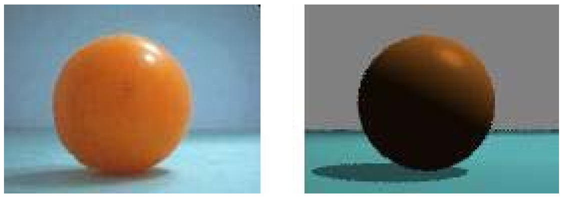


FIGURE 6. ObjetiveImage and RenderedImage.

In generation 201, the light source position had been found out by the algorithm, obtaining good values for X, Y and Z coordinates fig. 7. Values are from 0 to 1 and then are multiplied by 10 to obtain virtual meters. In the same way the light source intensity come across its best value. The intensity value is multiplied by 100 to obtain the parameter for the renderer, fig 8.

The Red, Green, and Blue sphere color components coma across at their best in generation 341, fig. 9. Then it found out the local best, due the limitations of the illumination model we use. We think that using a better model, we obtain better results.

The RGB plane color components found out their best at generation 341, fig. 10, then the local best, the same the sphere, due the limitations of the illumination model we use.

5. CONCLUSIONS

The proposed methodology, still in development, obtains good color and light source values. As we can see, it can improve image quality by adjusting the fitness function. As the image quality is directly dependent on fitness function, the basic implementation of the methodology gives good results. For example it is really surprising that using only these two pixels the algorithm finds a good position for the light source, as we can see in the shadow. In the same way the implemented algorithm finds good intensity value for light source, unless the object color can be improved. So we will continue working on this investigation to find better results.

As future work we are improving the illumination model to obtain better rendered images and programming the algorithm in a parallel architecture to improve speed.

REFRENCES:

[1] John P. Collomosse: “Supervised genetic search for parameter selection in painterly rendering”. Department of Computer Science, University of Bath, Bath, U.K.(2007)

[2] Furukawa R., Kawasaki H., I. K., and M, S.: “Appearance based object modeling using texture database: acquisition, compression and rendering”. ACM International Conference Proceeding Series 28\, 257--266, (2002)

[3] Gero Muller, Gerhard H. Bendels, R. K.: “Rapid synchronous acquisition of geometry and appearance of cultural heritage artefacts”. (2005)

[4] Wojciech Matusik, Hanspeter Pfister, R. Z. AN., and McMillan, L.: “Acquisition and rendering of transparent and refractive objects”, (2002)

[5] G. Muller, J. Meseth, M. S. R. S., and Klein, R.: “Acquisition, synthesis and rendering of bidirectional texture functions" In: EUROGRAPHICS 2004 State of The Art Report (2004)

[6] Ngan, W. K. A.: “Acquisition and Modeling of Material Appearance”. PhD thesis, Massachusetts Institute of Technology. (2004)

[7] Srinivasa G. Narasimhan, Mohit Gupta, C. D. R. R. S. K. N. H. J.: “Acquiring scattering properties of participating media by dilution”, ACM transactions on Graphics 25, 3, 1003--1012., (2006)

[8] Collomosse, J.P., Hall, P. M.: “Genetic paint: A search for salient paintings”. In: proc. EvoMUSART (at EuroGP), Springer LNCS. Volume 3449. pp. 437-447, (2005)

[9] Unemi, T.: SBART2.4: “Breeding 2D CG Images and Movies, and Creating a type of Collage”, In: The Third International Conference on Knowledgebased Intelligent Information Engineering Systems. Adelaide, Australia, 288-291(1999)

[10] Unemi, T.: SBART home page. http://www.intlab.soka.ac.jp/~unemi/sbart/ (2009)

[11] Todd, S., Latham, W.: “Evolutionary Art and Computers”. Academic Press, (1992)

[12] Rowbottom, A.: “Evolutionary art and form”, In: Bentley, P.J., ed.: Evolutionary Design by Computers. Morgan Kaufmann, 261-277 (1999).

[13] Todd, S., Latham, W.: “The mutation and growth of art by computers”. In: Bentley, P.J.,

ed.: Evolutionary Design by Computers. Morgan Kaufmann, 221-250, (1999)

[14] Dutre, P., Suykens, F., Willems, Y.: “Optimized Montecarlo Path Generation using genetic algorithms”, Katholieke Universiteit Leuven Department of Computer Science (1998)

[15] Beyer, M., and Lange, B.: “Rayvolution: An evolutionary ray tracing algorithm”. In: Eurographics Rendering Workshop 1994 Proceedings, pp. 137-146. Also in Photorealistic Rendering Techniques, Springer-Verlag, New York,(1995)

[16] Veach, E., Guibas, L. J.: “Metropolis Light Transport”, Computer Science Department Stanford University, (1990)

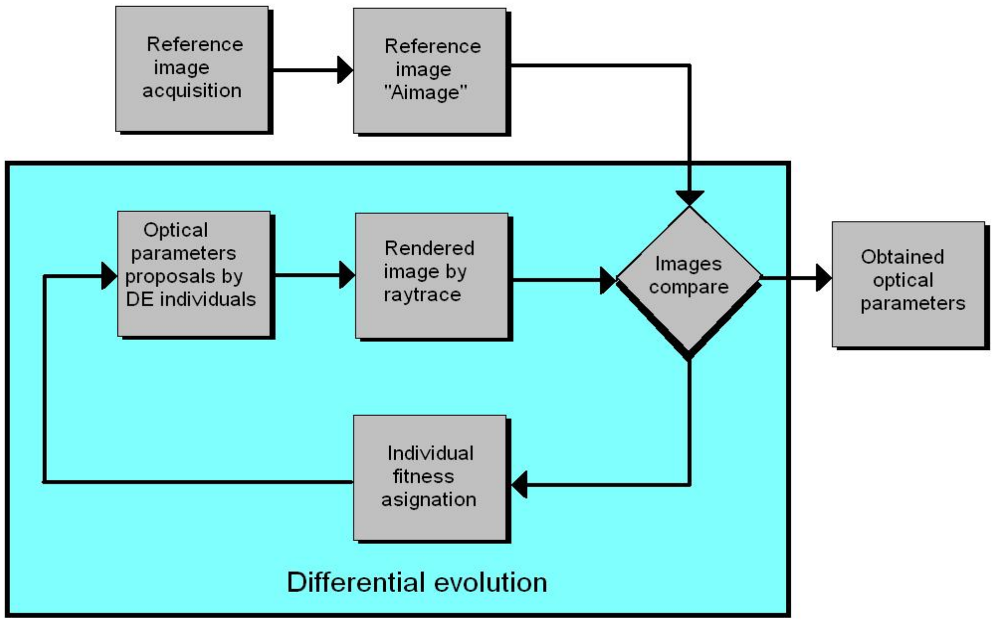

Reference image acquisition
Reference image "Aimage"
Optical parameters proposals by DE individuals
Rendered image by raytrace
Images compare
Obtained optical parameters
Individual fitness asignation
Differential evolution

FIGURE 1. Proposed methodology for rendering in the loop parameter extraction.

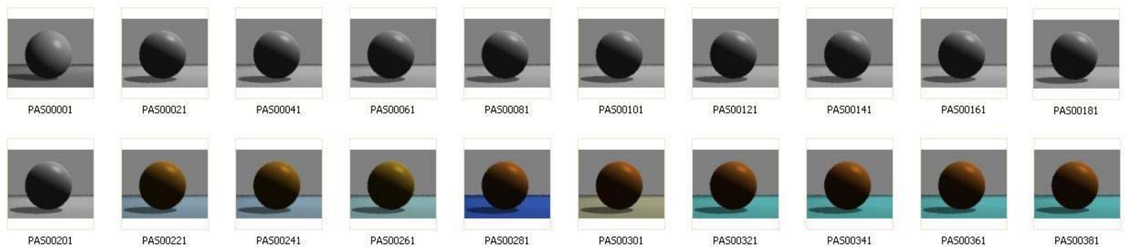


FIGURE 2.Sequence of generated images, where the first set, from PAS00001 to PAS000201, finds the light source position and intensity. And the second set, from PAS000221 to PAS00381, try to find the color for the floor and the sphere.

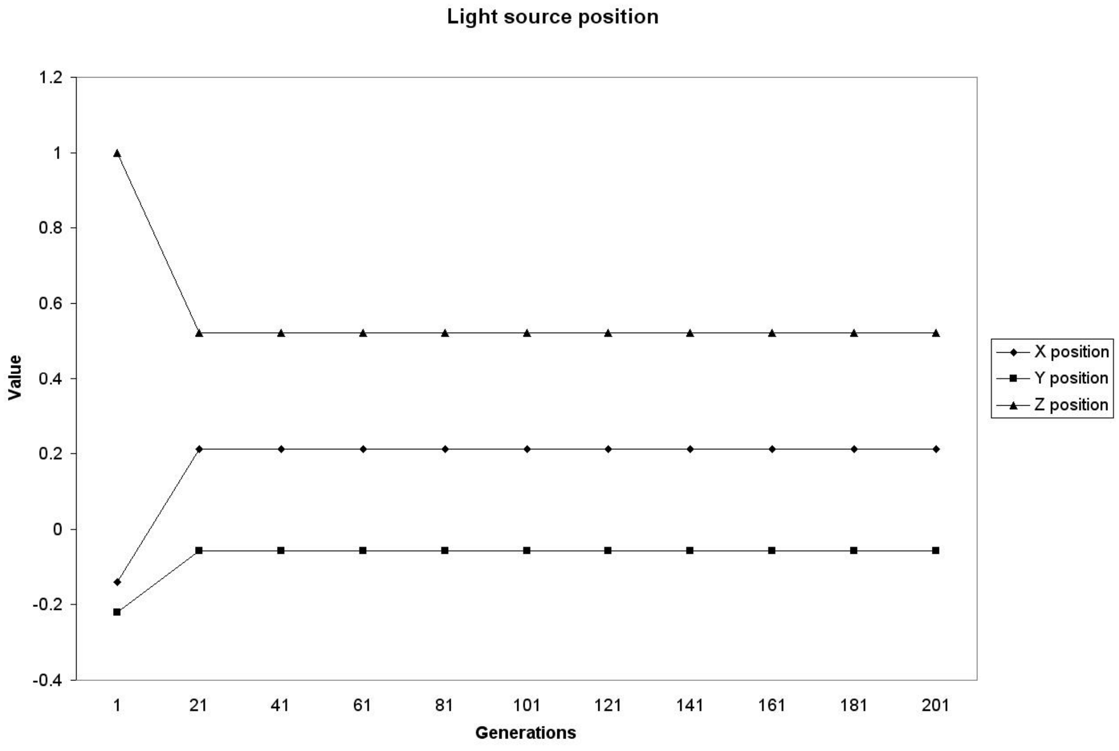


FIGURE 7. Light source position found out by the comparison function 1.

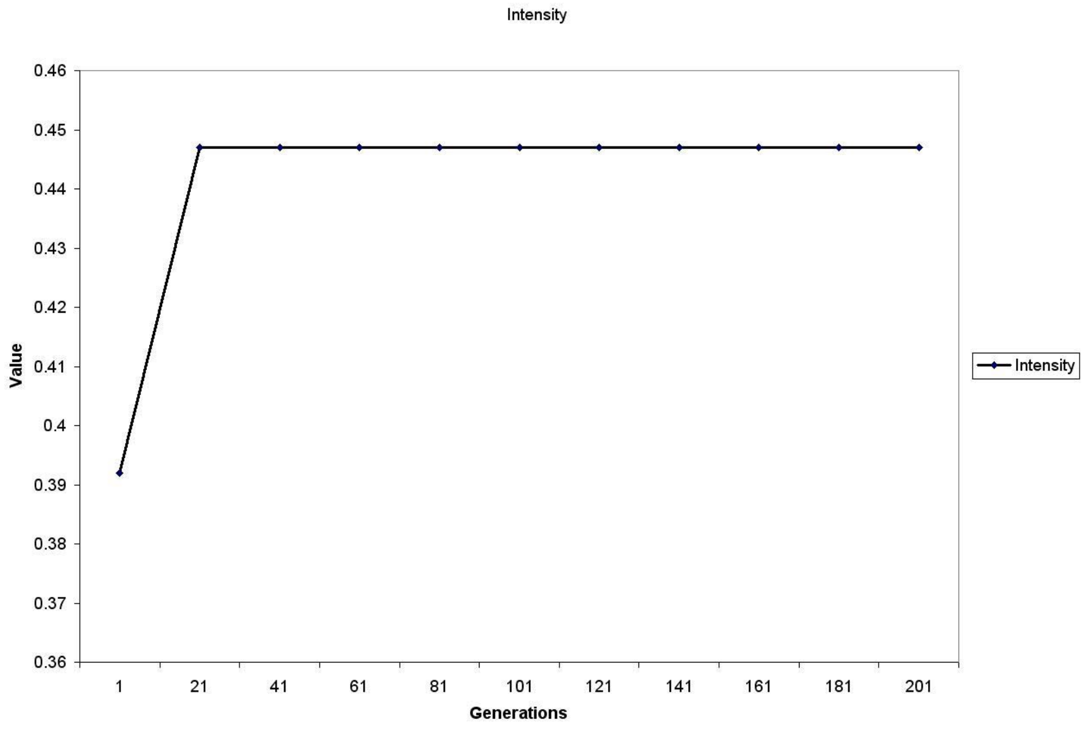


FIGURE 8. Light source intensity, found out by the algorithm.

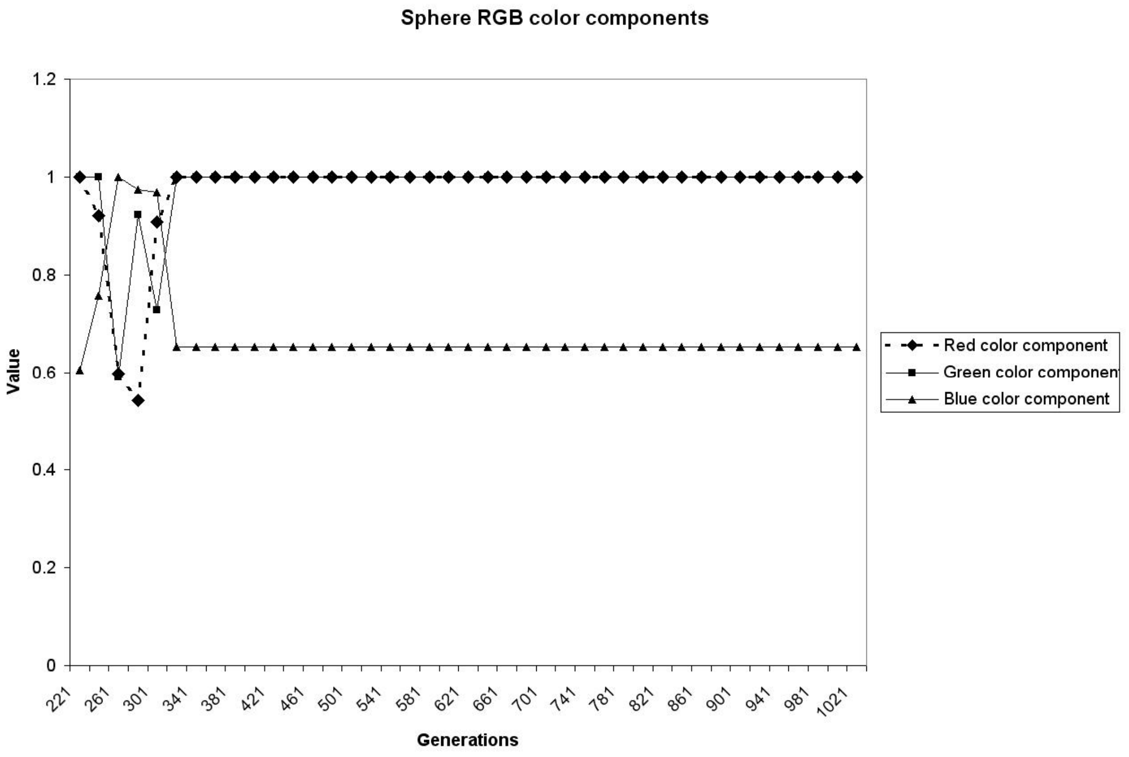


FIGURE 9.Evolved sphere RGB components.

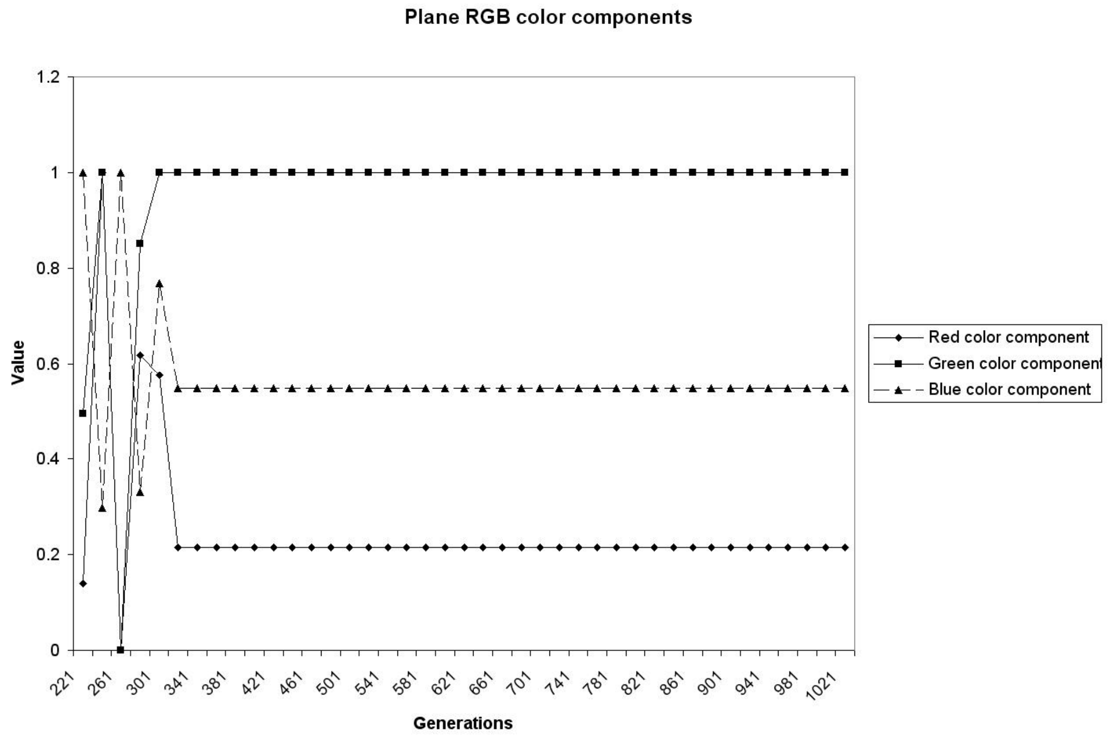


FIGURE 10. Evolved sphere RGB components.