Evolving Assemblages of 3D Objects

Fernando da Graça and Penousal Machado
CISUC, Department of Informatics Engineering, University of Coimbra
Polo II of the University of Coimbra, 3030 Coimbra, Portugal
fejg@student.dei.uc.pt, machado@dei.uc.pt

Abstract

A novel approach to the creation of assemblages of three-dimensional digital objects is presented and explored. The proposed evolutionary art approach allows the evolution of the distribution of 3D objects, which are placed on a virtual canvas, constructing a non-photorealistic transformation of a source image. The approach is thoroughly described, giving particular emphasis to the interaction between the artist and the tool, and to influence of the artistic decision making process in the final outcome. The experimental results presented highlight the differences between several styles of images, evolved in accordance to different artistic purposes, showing the potential of the approach for the production of large-scale artworks.

Key words: Digital Art, Evolutionary Art, 3D Assemblages

1. Introduction

The main goal of the research presented in this paper is the creation of an interactive evolutionary art tool for the creation of large-scale assemblages of 3D digital objects. Being an interactive tool, we are particularly interested in ensuring that the users are able to convey their artistic preferences and ideas through it. That is, the users should be able to significantly influence the final outcome of the process, express themselves through the use of this tool, and recognize their *signature* in the evolved artworks, thus connecting with them at an emotional and artistic level.

The main artistic sources of inspiration for this work are pointillism, mixed media assemblage of objects, and ornamentation techniques (e.g. similar to the ones found in Gustav Klimt works). From a scientific point of view, areas such as evolutionary non-photorealistic rendering and artistic filter evolution, are of particular relevance.

Synthetically, the evolutionary process leading to the creation of an assemblage of 3D digital objects can be described as follows:

(i) The user provides an input source image (e.g. a photograph);

(ii) The user creates, or selects, a library of 3D objects that will be used to create the assemblage;

(iii) Through a user-guided evolutionary process the user evolves the rotation, size and placement of the objects:

   (a) An initial random population of object assemblages is created;

   (b) The user indicates those that better match his/hers ideas;

   (c) The next population is created through the crossover and mutation of the genetic code of the selected assemblages;

   (d) The process is repeated from step (b) until the an assemblage that satisfies the preferances of the user is found.

(iv) The 3D scene, which constitutes a non-photorealistic portrayal of the source image is rendered using a raytracer;

(v) Depending on the nature of the employed objects and on the complexity of the scene the artist may chose to physically replicate the virtual assemblage.

During the evolutionary stage, the user explores different object distributions. The placement, size and rotation of the objects have a considerable impact on the final outcome.

The distribution of objects can help to saturate a deceptive movement and give rhythm to the final work. On the other hand, it can also completely distort the input image. The nature of each object is preserved, i.e. the shape and materials used in each object are not altered by the evolutionary process. However, the color of each object is determined by the color of the corresponding pixels of the input source image.

The proposed approach can be seen as an instance of

Computer-Aided Creativity [1], in the sense that the tool takes care of several aspects related with the technical execution of the artwork, allowing the user to focus on the creative aspects of the task, and provides mechanisms that allow the exploration of a search space of potential artworks, guided by the artistic and aesthetic preferences of the user. In this way, the artwork arises from the interaction between user and tool. Due to the stochastic nature of the process, serendipity plays an important role, sometimes leading the artist to explore unforseen paths and diverting the artist from his/hers original ideas. On the other hand, user guided-breeding promotes the recombination of the individuals that are closer to the aesthetic preferences of the user, leading to the successive refinement of the populations.

We begin with a short introduction to evolutionary art. In the third section, we make an overview of the different modules of the system. In the fourth section, we describe the evolutionary process. The experimental results are presented and analyzed in the fifth section. Finally, we draw some conclusions and discuss aspects to be addressed in future work.

2. Evolutionary Art

Through time, natural evolution gave rise to a huge variety of species adapted to their environment. The diversity of their shapes, and of the ways they are able to survive, interact and change their environments, allows us to claim that nature is creative, or, if we prefer to see it this way, the mother of all creativity. From a different perspective, natural evolution can also be regarded as an optimization process, i.e. finding the fittest species for a given environment.

From Holland's work [2] onwards, natural evolution has also become the basis for several Artificial Intelligence approaches, usually referred to as Evolutionary Computation (EC). In general terms, EC approaches imitate the fundamental mechanisms of evolution: *selection*, which guarantees that the most apt individuals have greater probabilities to survive and reproduce; *reproduction*, which ensures the inheritance of the parents' genetic material, as well as *variation*, which allows evolution. Thus, EC transforms Darwin's ideas into algorithms, allowing the evolution of populations of solutions for specific problems [2].

Currently, there are four main EC approaches: Evolutionary Programming [3], Evolution Strategies [4], Genetic Algorithms [2], and Genetic Programming [5]. All of them have been successfully applied to a large variety of problems, mainly optimization ones.

The idea of using EC for artistic purposes can be traced back to Richard Dawkins [6], who developed a simple genetic algorithm that allowed the evolution of the shapes of virtual organisms called "biomorphs". One of the key ingredients of this approach was user interaction: the user evaluated the individuals of the population according to his/hers preferences. Those which were better classified had higher probabilities of generating offspring. As a consequence, the populations gradually become closer to the user's preferences. This technique of artifact generation has been named interactive evolutionary computation.

Following the same set of ideas, Karl Sims used Genetic Programming to evolve populations of images. In Genetic Programming the genetic code of each individual (genotype) is a program, in Sims' case a symbolic expression that once rendered becomes an image (phenotype). The user assigns fitness to the images, thus indirectly determining the survival and mating probabilities of the individuals. The fittest individuals have a higher probability of being selected for the creation of the next population, which is generated through the recombination and mutation of the genetic code of the selected individuals.

The seminal work of Karl Sims allowed the evolution of striking imagery, and the success of his system lead to the application of evolutionary computation approaches to several artistic tasks, including: image generation, animation, sculpture, architecture and design (a thorough survey can be found in [7]).

The use of evolutionary algorithms to create image filters and non-photorealistic renderings of source images has been explored by several researchers. Focusing on the works where there was an artistic goal, we can mention the research of: Neufeld and Ross [8,9], where Genetic Programming (GP) [5], multi-objective optimization techniques, and an empirical model of aesthetics are used to automatically evolve image filters; Lewis [10], which evolved live-video processing filters through interactive evolution; Machado et al. [11], where GP is used to evolve image coloring filters from a set of examples; Yip [12], which employs Genetic Algorithms (GAs) to evolve filters that produce images that match certain features of a target image; Collomosse [13,14], which uses image salience metrics to determine the level detail for portions of the image, and GAs to search for painterly renderings that match the desired salience maps. Several other examples exist, however a thorough survey is beyond the scope of this article.

3. Overview of the System

Figure 1 presents the architecture of the system, which is composed of two main components: an *Evolutionary* module and a *Previewing and Rendering* module.

The evolutionary module is an expression–based GP [15] interactive breeding tool, that is responsible for the evolution of a population of assemblages. The user selects two parents, which generate offsprings through *crossover* and *mutation.* Crossover recombines the genetic code of the parents, being therefore responsible for the exploitation of characteristics that are already present in the current population. The mutation operator induces small changes in the genetic code, promoting exploration. An analogy can be drawn between this process and some aspects of the artistic

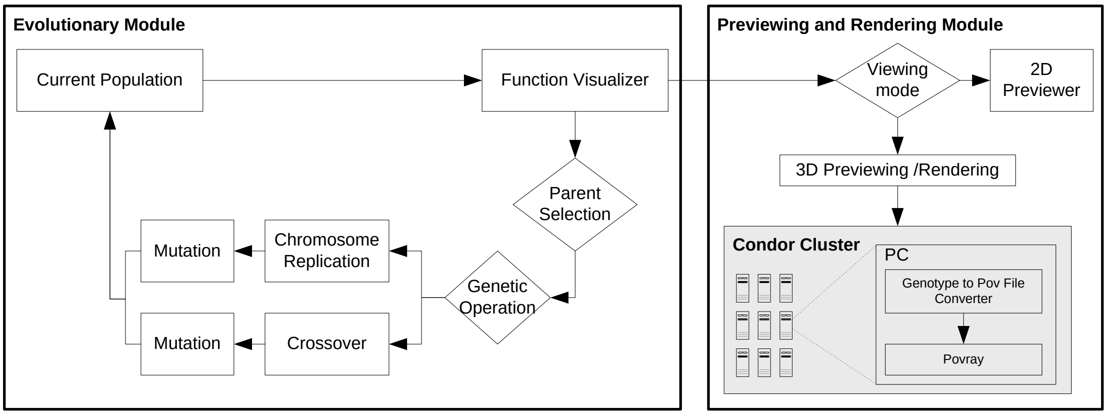


Fig. 1. The main modules of the system.

production of several artists. For instance Francis Bacon was known to preserve and select fortuitous "accidents", creating in this way a painting that was different from the originally envisioned one. In a similar way, the users preserves and select variations of the individuals, evolving images that are match their preferences. Although randomness plays an important role, as evolution progresses the choices of the user are continuously steering the algorithm to a particular style of imagery, making the images increasingly refined and unique.

To see the generated assemblages the user can use the 2D or 3D previewer. The 2D previewer runs on the master computer. It evaluates the genotypes and places objects accordingly. However, as the name indicates, it doesn't take into consideration the 3D nature of the objects, lighting effects, shadows, etc. The 3D previewer employs a render farm to produce a 3D raytraced rendering of the scene. For that purpose it uses POV-Ray [1] , a open source raytracer.

4. Evolutionary Process

In this section we describe the evolutionary module, focusing on aspects such as: representation, genetic operators and genotype–phenotype mapping.

4.1. *Representation*

The genotype of each individual has five chromosomes: <*type, rotation, size, x-position, y-position*>. Each chromosome is an expression tree, encoding a particular aspect of the 3D assemblage of objects, as follows:

<***type***> – The output value of the *type* expression tree determines what object, from a pool of available ones, will be placed;

<***rotation***> – Determines the rotation that will be applied to the object;

<***size***> – Determines the scaling applied to the object.

<***x-position***> – Determines the x coordinate where the object will be placed;

<***y-position***> – Determines the y coordinate where the object will be placed.

The internal nodes of the tree are functions from the following set:

$$\{sin, cos, max, min, abs, +, -, \times, \%, diff\},$$

where *sin* and *cos* are the usual trigonometric operations; *max* and *min* are functions that take two arguments returning, respectively, the maximum and minimum value; *abs* returns the absolute value; $\{+, -, \times\}$ are the standard arithmetic operations; % the protected division operator [5]; $diff$ a function that returns the difference between the current pixel and the one specified by the coordinates passed by the arguments of the function.

The leaf nodes are terminals. These can be variables (x, y), random floating constants, or the source input image.

4.2. *Genetic Operators*

Three genetic operators are used: crossover, mutation and chromosome replication.

The crossover operator is based on the standard GP subtree exchange crossover [5]. It randomly selects a sub-tree from each parent and exchanges them, creating two descendentes. This operator is applied to each homologous chromosome pair, meaning it can only exchange subtrees between similar chromosomes (i.e. a subtree of the chromosome encoding size cannot be replaced by a subtree of a chromosome encoding rotation). For each homologous chromosome pair there is a probability of occurrence of crossover.

The mutation operator randomly selects a subtree and replaces it by a randomly created one.

The chromosome replication operator was introduced to propagate a specific chromosome throughout the entire population, thus allowing the user to test it in different conditions. E.g., the user may feel particularly pleased with the rotations applied to the objects in one individual, and

[1] http://www.povray.org/.

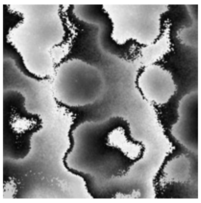
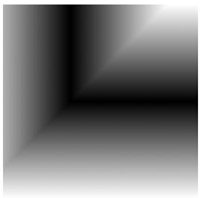
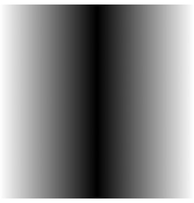
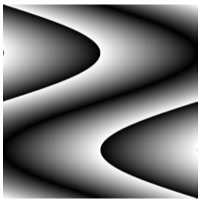
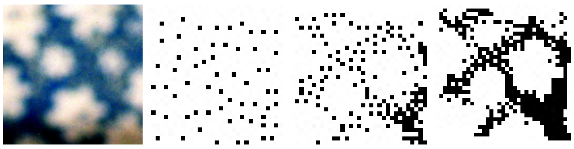
Type	Rotation	Size	X-position	Y-position
max(1.79,+(image,x))	min(x,-(1.8, sin(max(y,1.9))))	min(y,-(min (x,x),sin(max (x,1.9))))	abs(x)	-(sin(y),x)
				

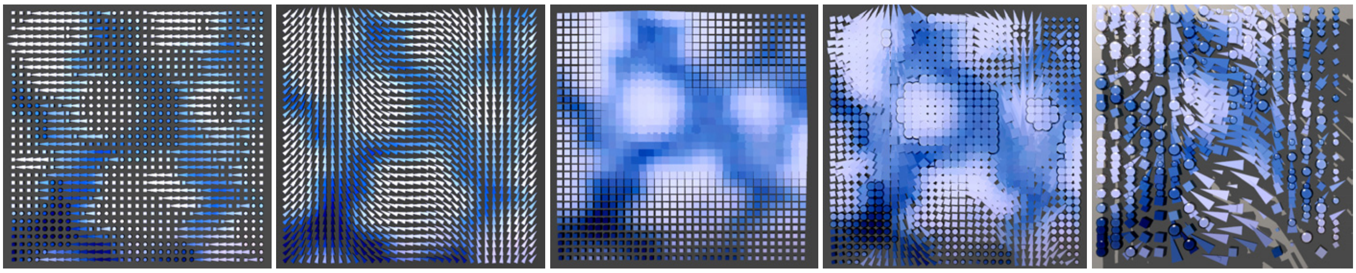
Fig. 2. Chromosomes of a sample genotype and the visualization of the corresponding functions over $[-1, 1]$, considering the source image presented in Fig. 3(a)

.



(a) (b) (c) (d)

Fig. 3. (a) Source Image; (b) to (d) Three dither masks.



(a) (b) (c) (d) (e)

Fig. 4. Assemblages resulting from the application of *type* (a), *rotation* (b), *size* (c), and <*type, rotation, size*> (d) to the source image of Figure 3(a), assuming a regular grid placement of the objects. Assemblage (e) is the result of simultaneously applying <*type, scale, rotation, x, y*>.

wish to use the same rotation expression in all individuals. Alternatively, the user may wish to test small variations of a specific chromosome without changing the remaining ones. To address these needs, the chromosome replication (a) copies the chromosome selected by the user to all individuals in the population, replacing the corresponding ones, (b) applies, in each individual, a node–change mutation to the copied chromosome. When applied to a random constant *node-change* mutation introduces a Gaussian perturbation. When applied to other node types, it replaces the function or terminal by a different one, creating or deleting subtrees when the arity of the functions is different.

4.3. *Genotype–Phenotype Mapping*

In this section we describe how the genetic code of an individual is transformed in a assemblage of 3D objects.

To ilustraste our explanation we resort to the genotype presented in Fig. 2 and the source image presented in Fig. 3(a). For the time being we will assume that the objects are placed following a regular 32×32 grid, and that three types of objects are available: cubes, spheres and pyramids.

The first chromosome, *type*, determines which type of object is placed. In this case, values in $]0, 0.33]$ correspond to cubes, in $[0.33, 0.66]$ to spheres, and in $[0.66, 1[$ to pyramids. The application of the *type* chromosome of Fig. 2, alone, would produce the 3D scene depicted in Fig. 4(a).

The *rotation* chromosome determines the rotation that will be applied to each object. The application of the *rotation* chromosome of Fig. 2 would result in the 3D scene presented in Fig. 4(b), where we used pyramids for easier viewing of the effects of rotation.

Likewise, *size* determines the scaling that will be applied to the object. Figure 4(c) depicts the results of independently applying the *size* chromosome of Fig. 2 using cubes.

So far we considered that the objects are placed on a reg-

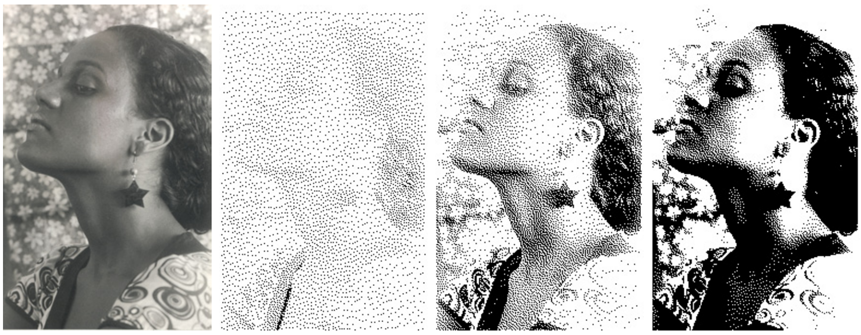


Fig. 5. Source image and corresponding dither masks.

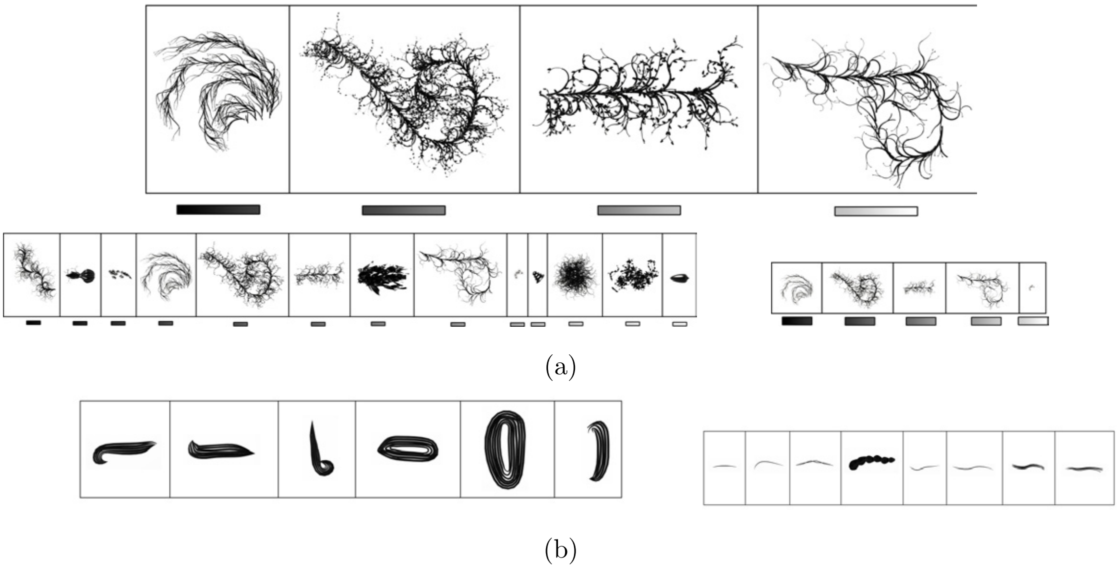


Fig. 6. Object libraries used in the experiments.

ular grid. This type of placement has characteristics that we wish to avoid, namely: i) the regularity of the grid can become a visual distraction; ii) it only allows a homogeneous distribution of the objects, making it impossible to ignore regions of the image, or to clutter objects on certain regions.

To overcome this limitation we introduced the *x-* and *y-position* chromosomes, which determine the coordinates where the objets are placed (see Fig. 4e).

The number of objects placed is also relevant. To address this issue we resort to masks. A modified version of a space-filling curve dither algorithm [16,17] is applied to the source image. By establishing different parameter settings, one can create different dither masks (see Fig. 3).

The phenotype is then rendered in several stages, each using a different dither mask. In each stage, the positions of the objects are calculated using the *x-* and *y-position* chromosomes, but an objects is only placed if the mask allows it.

The masks allow an additional degree of control to the artist. For instance, he may chose to create different masks for different regions of the image, thus indirectly determining de level of detail of different areas, which can be particularly useful to highlight salient detail or to abstract areas of less importance.

5. Experimentation

The analysis of the experimental results attained by evolutionary art systems, specially user driven ones, entails a high degree of subjectivity. In our case, there is an additional difficulty: our approach is thought for large-scale formats, therefore is close to impossible to adequately convey the real look of the evolved images in the space and format available for their presentation.

Considering these difficulties we focus on the presentation of assemblages created by our approach, highlighting the influence of the artistic decision making in the final outcomes.

5.1. *Experimental Setup*

We used the following experimental settings: Function–set = $\{sin,\ cos,\ max,\ min,\ abs,\ +,\ -,\ \times,\ \%,\ diff\}$; Terminal–set = $\{x,\ y,\ image,\ random_{constants}\}$ (see Sect. 4.1); Population size = 20; Number of Generations = 40; Crossover probability = 0.6; random–subtree mutation probability = 0.2; node–change mutation probability = 0.02 per node; Population initialization method = Ramped half–and–half.

The experiments were performed on an Intel Core2Duo, 2.8GHz, Windows master computer. During the course of evolution, an heterogeneous computer cluster was used for the 3D previewing of the populations. Depending on the moment in time 40-70 machines were available. To produce large-scale renderings, we used a dedicated cluster with networked file system, composed of 24 Intel Core2Duo, 2.8GHz, running Ubuntu. For previewing we used a resolution of 800 × 600 pixels, and the images were rendered without anti-aliasing, the resolution of the large-scale renderings ranged from 3200 × 2400 to 16000 × 12000.

5.2. *Results and Analysis*

To assess the strengths and weaknesses of our approach we used different types of input image, namely "still" face photographs, upper body photographs that suggest movement, and landscapes. In Fig. 5 we present one of the source images used in these experiments, and the corresponding dither masks using during rendering.

The creation of the libraries of objects followed the following general rule: Include objects of varying size and complexity. The rationale is that large objects could be used to create the raw shape of the input image, while small and thin objects could be used to provide detail to the areas of the image that we wished to emphasize. Figure 6 depicts the two different object libraries used in the course of the experiments.

The first object library (Fig. 6a) was composed, mainly, by branches, leafs and other botanical inspired shapes. With the second object library we which to attain assemblages with an appearance that is evocative of tapestry and woven baskets. For this purpose we employed simpler objects that, when assembled, could produce the interweaved appearance that we aimed for. In addition, we created several objects that resemble strands to provide detail to areas of the image that we might want to reproduce closer.

In the first stages of evolution our posture was mainly exploratory, in the sense that we welcomed individuals that differed in style from the ones that we had already saw, thus valuing novelty. In later stages, namely once a promising assemblage was found, we exerted an higher degree of guidance, forcing the algorithm to focus on variations of this assemblage. In this way, we promoted the successive refinement of the promising assemblage transforming it in one that better matched our preferences.

In Fig. 7 we present several examples of images evolved using the library of botanical shapes. As it can be observed the choices made by the user during the evolution have a significant impact in the final outcome.

In the evolutionary process that generated the assemblage presented in Fig. 7a we wished to attain an abstract sand-like effect. This was eventually achieved by an individual that places a vast number of objects, consequently causing high degree of overlap among them, which makes individual objects invisible and produces the effect we were seeking. Figure 7b is the result of serendipity: during the course of evolution a fortuitous mutation created an abstract assemblage with an interesting contour. This image was later refined during the course of several generations leading to the creation of the presented assemblage.

In the evolution of the assemblages presented in Fig. 7a and Fig. 7b we valued the individuals that better captured the expression of the source image, giving particular emphasis to the eyes.

In Fig. 8 we present several assemblages evolved using the second object library (see Fig. 6b). As it can be observed by the comparison of these assemblages with the ones of Fig. 7, a change in the object library can have a dramatic effect on the final outcome.

During the evolution of these assemblages we gave particular emphasis to several different aspects, including: the rendering of the eyes (Fig. 8a, see detail); conveying an sense of motion (Fig. 8b), abstracting detail and mimicking texture (Fig. 8c); overall expressiveness (Fig. 8d).

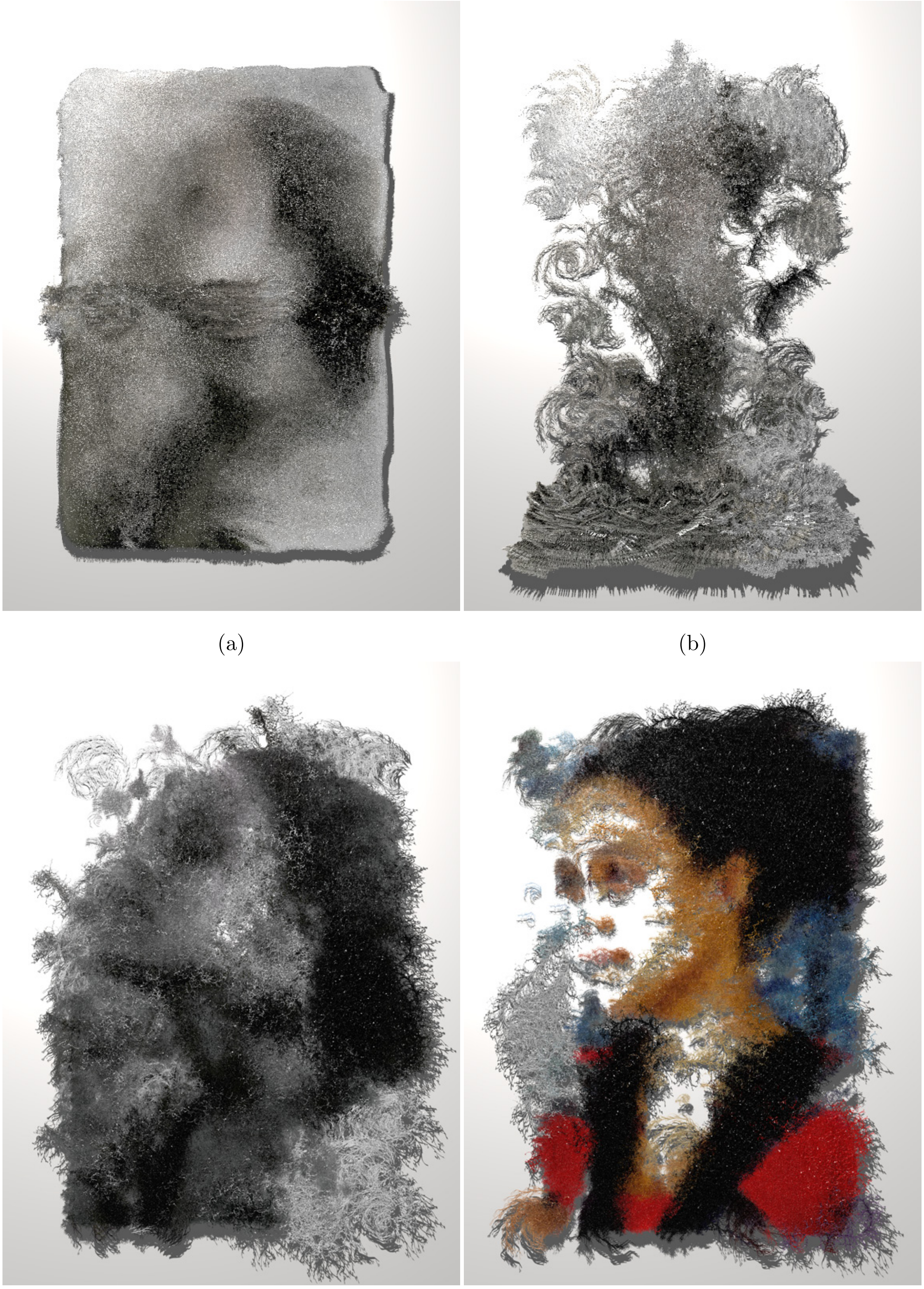
Its interesting to notice that although the objects are placed individually on the virtual canvas, there is a sense of continuity which results, mostly, from the artificial lines created by the regularity in object alignment and rotation. This effect can be better observed in the neck region of Figs. 8a and 9a and on the background of Fig. 8b.

Although the objects are three-dimensional, they are placed in a two-dimensional plane. Therefore, the sense of volume and depth present in the assemblages (e.g. Fig. 8b) results from matching the colors of the objects to those of the original photograph. For the creation of the assemblage presented in Fig. 9a, we explored a variation of the previously described evolutionary technique. In this case the assemblage was evolved in two stages: in the first an assemblage for the face photograph was evolved; in the second we created an assemblage for the red earring. These two assemblages were merged, with the objects reproducing the earring placed closer to the viewpoint.

In Figs. 9b and 9c we present two additional images, created with an object library composed exclusively by spheres.

6. Conclusions and Future Work

We described a novel evolutionary approach for the generation of assemblages of 3D digital objects, presenting results attained with different source images and object libraries. The experimental results show that the initial choices done by the user – selection of source image and of object library – have a significant impact in the final outcome. In addition, the interactive evolutionary process allows the users to explore the search space of assemblages, and to guide evolution in accordance to their artistic intent, eventually finding regions of the space that match his/hers



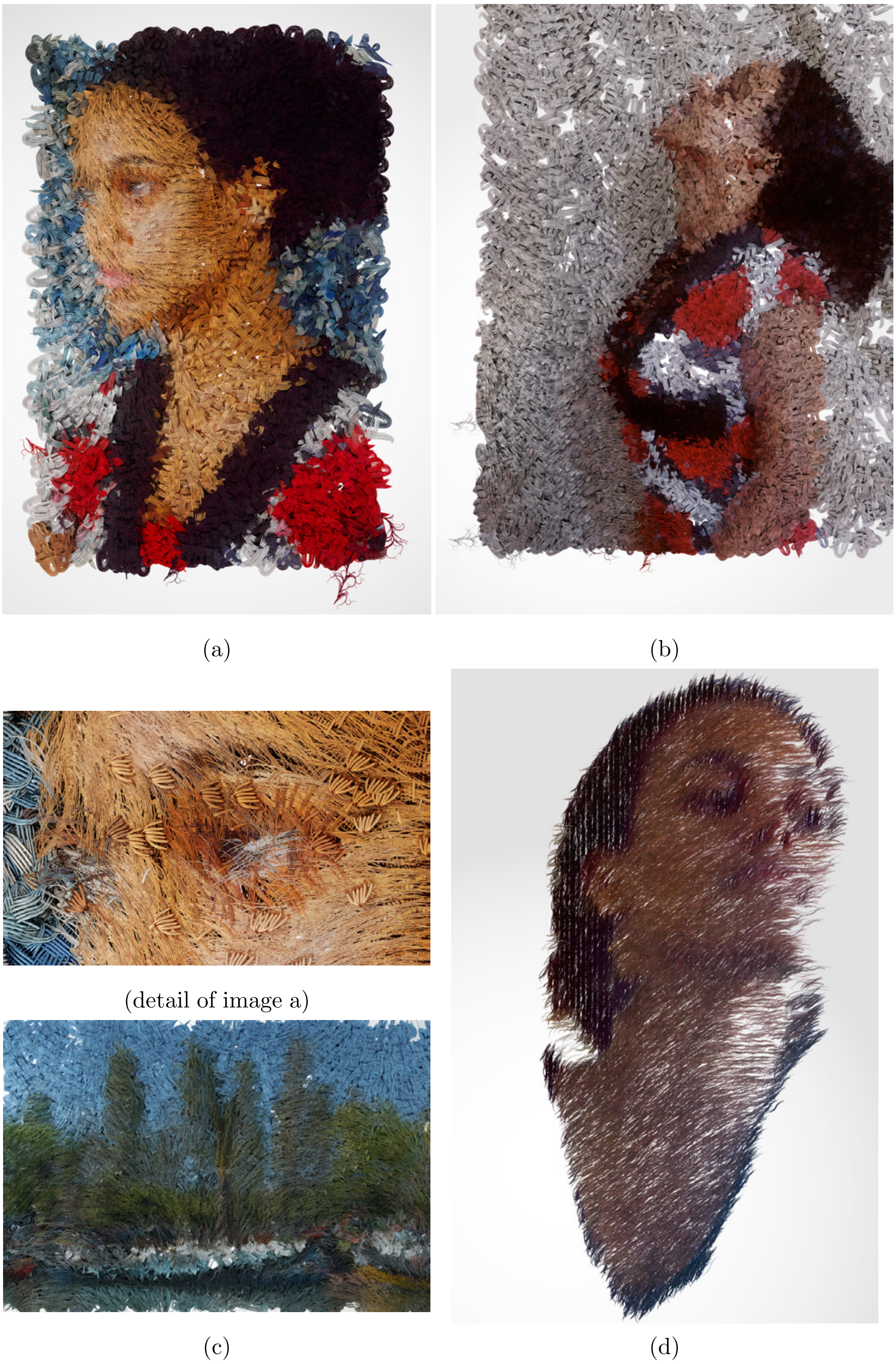
(a)

(b)

(c)

(d)

Fig. 7. Samples of images evolved using the object library presented in Fig. 6a.



(a)

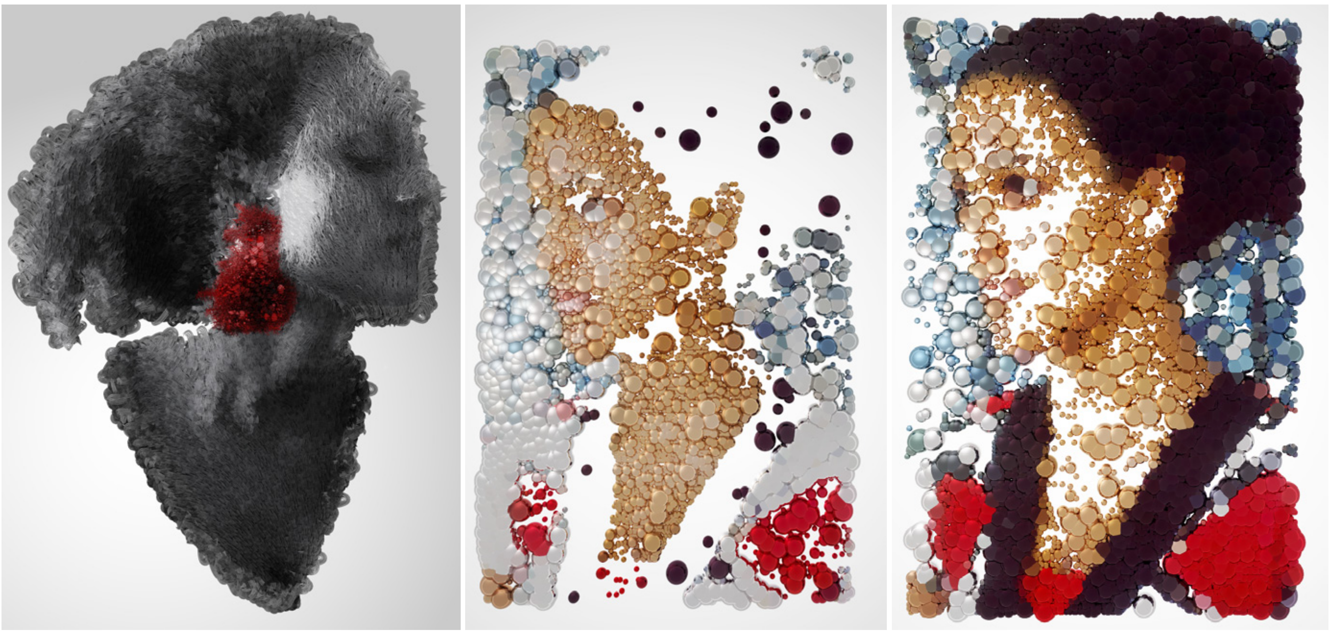
(b)

(detail of image a)

(c)

(d)

Fig. 8. Samples of images evolved using the object library presented in Fig. 6b.



(a)

(b)

(c)

Fig. 9. Some additional results.

preferences. In this way, the users are able to recognize their artistic signature in the evolved assemblages and to develop a sense of authorship of the evolved artworks.

One of the main limitations of our approach is the computational effort required to preview and render the individuals. To overcome it, used a computer cluster to distribute the rendering tasks. Nevertheless, this process is still time consuming, which becomes a limitation (specially for previewing since final renderings can be made offline). In the experiments presented we used fairly detailed objects. Using simplified versions of the same objects would greatly reduce previewing time. This is one one of the aspects that will be addressed in the future.

Although the object placement strategies we explored were able to produce interesting results, evolving the masks, would allow greater flexibility. Additionally, image salience analysis can play an important role. The identification of salient detail could be used to guide object placement, in order to promote the placement of objects in areas with salient detail.

Finally, a larger set of experiments is also necessary to better assess the strengths and weakness of the approach.

References

[1] Machado, P., Romero, J., Santos, A., Cardosoa, A., Pazos, A.: On the development of evolutionary artificial artists. Computers & Graphics **31** (2007) 818–826

[2] Holland, J.H.: Adpatation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)

[3] Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley & Sons, New York (1966)

[4] Schwefel, H.P.: Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik. Dipl.Ing. Thesis, Technical University of Berlin, Hermann Föttinger–Institute for Hydrodynamics (1965)

[5] Koza, J.R.: Genetic Programming: On the Programming of Computers by Natural Selection. MIT Press, Cambridge, MA (1992)

[6] Dawkins, R.: The Selfish Gene. Oxford University Press, Oxford (1976)

[7] Lewis, M.: Evolutionary visual art and design. In Romero, J., Machado, P., eds.: The Art of Artificicial Evolution: a Handbook on Evolutionary Art and Music. Springer (2007)

[8] Ross, B.J., Ralph, W., Hai, Z.: Evolutionary image synthesis using a model of aesthetics. In Yen, G.G., Lucas, S.M., Fogel, G., Kendall, G., Salomon, R., Zhang, B.T., Coello, C.A.C., Runarsson, T.P., eds.: Proceedings of the 2006 IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, IEEE Press (2006) 1087–1094

[9] Neufeld, C., Ross, B., Ralph, W.: The evolution of artistic filters. In Romero, J., Machado, P., eds.: The Art of Artificial Evolution. Springer (2007 (In Press))

[10] Lewis, M.: Aesthetic video filter evolution in an interactive real-time framework. In: Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC. Volume 3005 of LNCS., Coimbra, Portugal, Springer Verlag (2004) 409–418

[11] Machado, P., Dias, A., Cardoso, A.: Learning to colour greyscale images. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour – AISB Journal **1** (2002) 209–219

[12] Yip, C.: Evolving Image Filters. Master's thesis, Imperial College of Science, Technology, and Medicine (2004)

[13] Collomosse, J.P., Hall, P.M.: Genetic paint: A search for salient paintings. In: Applications of Evolutionary Computing, EvoWorkshops 2005, Lausanne, Switzerland (2005) 437–447

[14] Collomosse, J.P.: Supervised genetic search for parameter selection in painterly rendering. In: Applications of Evolutionary Computing, EvoWorkshops 2006, Budapest, Hungary (2006) 599–610

[15] Sims, K.: Artificial evolution for computer graphics. ACM Computer Graphics **25** (1991) 319–328

[16] Velho, L., de Miranda Gomes, J.: Digital halftoning with space filling curves. SIGGRAPH Comput. Graph. **25** (1991) 81–90

[17] Shiraishi, M., Yamaguchi, Y.: An algorithm for automatic painterly rendering based on local source image approximation. In: NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering, New York, NY, USA, ACM (2000) 53–58