Evolutionary Art using Processing

P. García-Sánchez, J.J. Merelo
Dept. of Computer Architecture and
Computer Technology
University of Granada
Email: pgarcia@atc.ugr.es

D. Calabria
Proenium
Email: dcalandria@proenium.com

A. B. Pelegrina
Department of Translation and Interpreting
University of Granada
Email: abpelegrina@ugr.es

R. Morcillo, F. Palacio
ETS. Informática y Telecomunicación
University of Granada
Email:robermorji@gmail.com, fruela@correo.ugr.es

R. H. García-Ortega
Fundación I+D del
Software Libre, Granada
Email: ruben@fidesol.org

***Abstract*—This paper shows how the Processing framework can be used to generate Evolutionary Art. Processing is a framework designed for visual designers and artists that is starting to have great presence in the interactive and visual art field. It includes several modules for image creation, manipulation and analysis. This framework has been used to create an Evolutionary Algorithm that generates abstract images comparing the histograms of a test image. Three different fitness functions have been used: the differences of the RGB histogram, the HSV histogram and an average of both. Results shows that the HSV also increases the similarities of the RGB (and therefore, the average) than the other two measures.**

I. Introduction

Evolutionary Art [1] is a type of generative art created by a computer following the principle of the survival of the fittest, using the methods from Evolutionary Computation [2]. A population of artistic works (individuals) are evaluated with an aesthetic measure to give a score (fitness). These individuals are combined and mutated to generate an offspring with inherited properties of the parents, during a certain number of times.

The main goal of this paper is to test the advantages of using Processing, a programming framework designed for visual artists, in Evolutionary Art. In this work Processing is used inside an Evolutionary Algorithm (EA) to model the individuals, generate their associate images and extract information of them (HSV, RGB and Average histograms) to fit with the histograms of a test image.

The rest of the work is organized as follows: in Section II, a brief review on Evolutionary Art is presented. Processing framework and image information are described next (Section III). The experimental setup and results are presented in Sections IV and V, respectively. Finally, the conclusions and future work can be found in Section VI.

II. State of the art

Computational Aesthetics "is the research of computational methods that can make applicable aesthetics decisions in a similar fashion as humans can" [3]. In the field of computational aesthetics, evolutionary systems can play an important role, enabling the evolution of aesthetically pleasing or innovative structures [4].

Representation of the art in Evolutionary Art can be one of the methods in the following classification:

- Symbolic expression. The genotype is a tree of expressions and the phenotype consists in the image produced by the evaluation of the tree.
- Grammars. A shape grammar is used as a formal description of the image.
- Using existing images as a source.
- Others, such as fractals or cellular automata.

One of the main challenges in Evolutionary Art is how to measure aesthetic value of an piece of evolutive art.

Two modes of aesthetics measures are defined by [5]:

1) "*Aesthetics evaluations are expected to simulate, predict or cater to humans notions of beauty and taste.*" This will be the definition used in this paper.
2) "*Is an aspect of meta-aesthetic exploration and usually involves aesthetic standards created by software agents in artificial worlds.*"

According to Galanter [5], computational aesthetics measures can be classified in the following categories:

- Based on Formulaic and Geometric Theories. The aesthetics of a piece of art are evaluated using a formula o principle (e.g., pythagorean proportions).
- Based in Design Principles. Like the rule of thirds or theory of color (e.g., using opposite colors).
- Based in Neural Networks and Connective Models.
- Complexity Based Models.
- Based in Evolutionary Systems:
  - Interactive Evolutionary Computation. The fitness of the individuals is determined by human agents.
  - Performance based goals. Certain properties of the art piece are evaluated and optimized based

in performance measures (e.g., usable surface in a furniture design generator).
  - ○ Error relative to Exemplars. The individual fitness is measured using a real-world example (e.g., a photography).
  - ○ Complexity measures. This type of measures is based in the idea the complexity is directly related to aesthetics and follows the path firstly stablished by Birkhoff [6].
  - ○ Multi-objective. Given the multidimensional nature of aesthetics judgement, multi-objective EAs are a clear option in order to deal with this multidimensionality.
  - ○ Extensions to EA (such as, coevolution, agent swarm behavior, etc.).

A brief classification of the aesthetic measures found in a short review can be found in Table I.

III. Processing and Histograms

In this section we will describe Processing [1]. Processing [15] is a framework formed by a simple programming language and an integrated development environment (IDE) mainly focused to electronic and visual artists, designers, musicians, etc.

Processing has the next advantages:

- Processing was created for artists, rather than programmers. So, it allows very complex drawings and interactive applications with few lines of code. For example, Figure 1(a) shows the sketch (in the IDE) necessary to create the Figure 1(b).
- It is an Open Source software (licensed under the GNU Lesser General Public License), and counts with a large development community.
- It is based in OpenGL, obtaining a good 3D acceleration.
- Includes more than 100 libraries for video, sound, physics, computer vision, networking, etc.
- Easy integration with Java, HTML5 and Android.
- Finally, it is fairly light when installed.

However, being a light framework, there exist some disadvantages:

- More complex applications require more programming skills.
- The calculations of large computer images are a bit inefficient (although expert programmers can manage OpenGL at low level to fix this).

There exist a lot of interactive artistic projects made with Processing; examples include art generation, artificial life, interactive music and other. A good selection can be seen in http://processing.org/exhibition/.

Processing is composed by several modules:

- Structure: Includes typical programming functions as is the case of return, draw (), void, and everything related to the structure of the program.
- Environment: Formed by the functions that handle the modeling of the window: cursor, width, height, background, for example.
- Data: formed by the data types that make up the different program variables (int, char, float).
- Control: Consisting in relations operators.
- Shape: it is formed by all functions of the treatment of figures 3D and 2D.
- Input: input interactivity features such as functions for the mouse, keyboard or files.
- Output: output interactivity features such as write on the screen, save the image or serial control.
- Transform: transformations such as rotations or translations.
- Lights and Cameras: Functions for the treatment of the lights and cameras.
- Color: Functions to handle color of the figures.
- Image: Functions for loading images or textures.
- Rendary: Functions for rendering images.
- Typography: Functions for dealing with text.
- Math: Functions for dealing with all mathematical functionality.

The Color module can be used to analyze images taking into account their histogram. The color histogram represents the frequency of occurrence of each color intensities present in the image, by accounting for such sharing pixels color intensity values.

The histogram is composed of different ranges or bins that represent a value or set of values of color intensity. The color space is defined as a model representation with respect to color intensity values: RGB (Red, Green, Blue) and HSV (Hue, Saturation and Value).HABLAR AQUI DE LA DIFERENCIA ENTRE HSV Y RGB. Figure 2 shows the RGB histogram of the image in the Figure 3.

IV. Experimental setup

This section show how Processing have been used in the EA, the individual representation, the fitness used, and the parameters of the experiments.

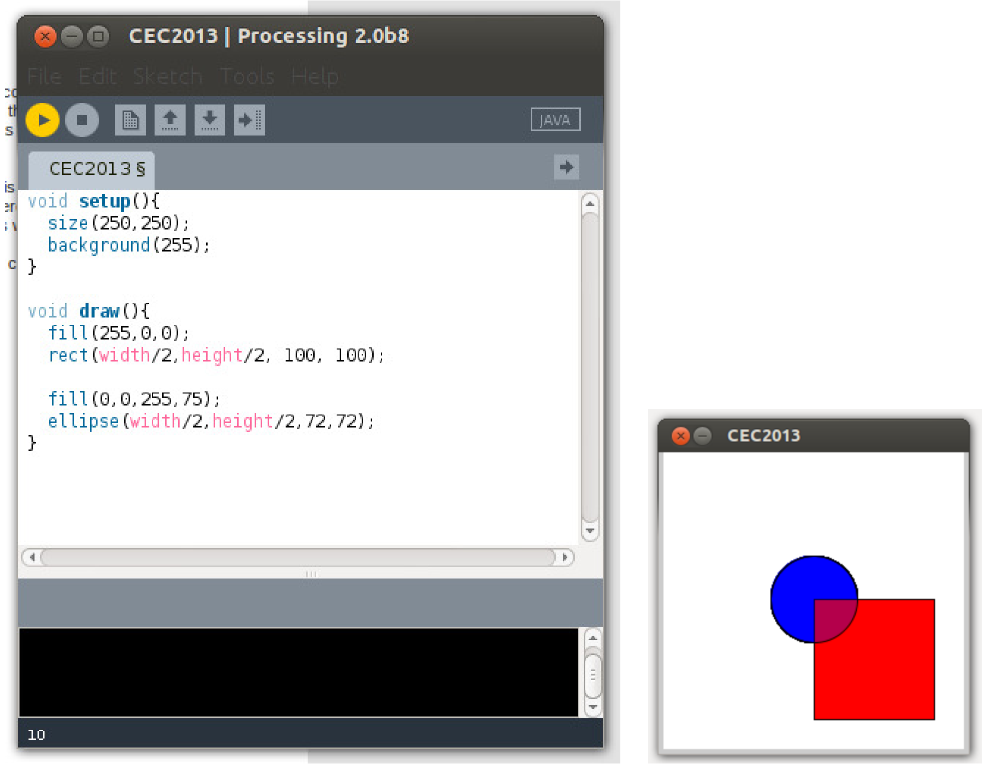
A. Integrating Processing in Java

Processing can be integrated with Java just adding a *jar* (a Java library) to existing software. The simplified code of the sketch (for example, the one shown in Figure 1(b)) is accessed extending the *PApplet* class. In this work, Processing has been integrated to an existent EA framework, OSGiLiath [16], a service-oriented framework based in Java that includes a lot of primitives and services for Evolutionary Computation. A new module called OSGiLiART has been added to the

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

TABLE I. CLASSIFICATION OF THE AESTHETIC MEASURES USED IN A BRIEF REVIEW OF THE LITERATURE ON EVOLUTIVE ART.

Type	Aesthetic Measure
Formulaic and Geometric Theories	Fractal dimension [7], Image order [8], Benford Law [9]
Based in Design Principles	Color contrast (hue) [10], Color ingredient [8], Composition, tonality and color [4].
Interactive Evolutionary Computation	The electric sheep project [11]
Error relative to Exemplars	Resemblance score [4], pixel comparation [12]
Performance based goals	Evolving virtual creatures [13]
Complexity measures	Image complexity [8], Machado and Cardoso aesthetic measure [14]



```
void setup(){
  size(250,250);
  background(255);
}

void draw(){
  fill(255,0,0);
  rect(width/2,height/2, 100, 100);

  fill(0,0,255,75);
  ellipse(width/2,height/2,72,72);
}
```

(a) Processing IDE

(b) Runtime of the code.

Fig. 1. Processing IDE and sketch execution.

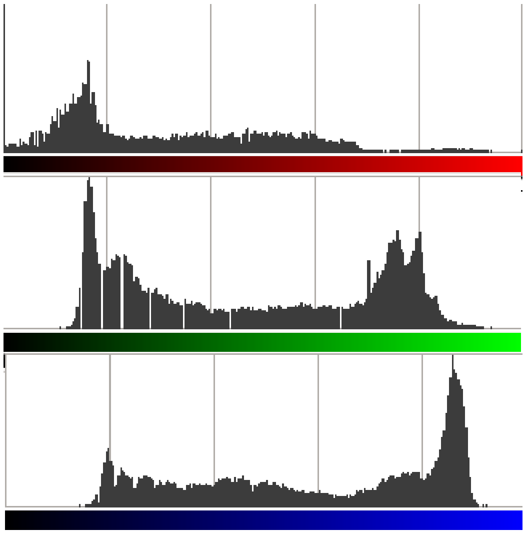


Fig. 2. RGB histogram of the Figure 3.

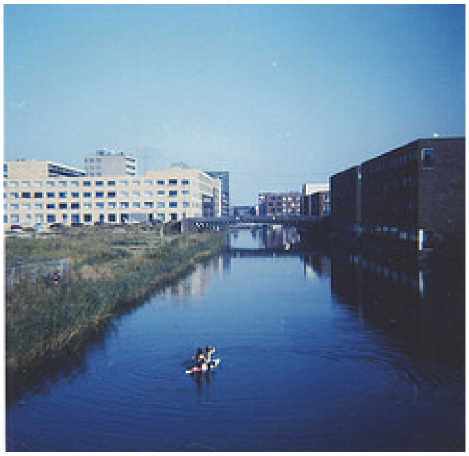


Fig. 3. Test image to compare with the Fitness functions of our algorithm.

publicly-available source code of OSGiLiath (available in http://www.osgiliath.org) under a GPL License. Then, using the packages available in the Processing library the EA can generate individuals, manipulate images or extract information.

B. Individual representation

To test the Processing advantages and perform the experiments, the genome of the individual is a list of Processing Circles. Each circle has a position, radium and color. This list can be recombined or mutated.

C. Fitness used

For this piece of research, we focused on two measures of aesthetics: basic histogram comparison. The fitness functions are included in the "Error relative to Exemplars" category, using Galanter [5] classification.

Three different fitness function has been tested:

- *RGB difference*: The difference of the RGB histogram of the individual with the RGB histogram of the test image.
- *HSV difference*: The difference of the HSV histogram of the individual with the RGB histogram of the test image.
- *Average difference*: An average of the two previous differences.

The range of the previous fitness have been normalized to vary from 0 (totally different histograms) to 1 (the same histogram).

For every color property (i.e., RED, GREEN, BLUE, HUE, SATURATION and VALUE), the histogram is computed using the expression (1) for each possible value (0-255). Then, again for every property, the difference between the target image and the individual histograms is obtained using (2). Finally, the three fitness are calculated: RGB fitness (6), HSV fitness (10) and AVERAGE fitness (11).

$$H(c, prop) = \frac{1}{N}\sum_{j=0}^{N}\begin{cases}1 & prop(j) = c\\ 0 & otherwise\end{cases} \quad (1)$$

$$diff(h_1, h_2) = \sum_{j=0}^{255} |h_1(j) - h_2(j)| \quad (2)$$

$$d_R(i) = diff(H(i, RED), H(target, RED)) \quad (3)$$

$$d_G(i) = diff(H(i, GREEN), H(target, GREEN)) \quad (4)$$

$$d_B(i) = diff(H(i, BLUE), H(target, BLUE)) \quad (5)$$

$$fitness_{RGB}(i) = 1 - 128\frac{d_R(i) + d_G(i) + d_B(i)}{3} \quad (6)$$

$$d_H(i) = diff(H(i, HUE), H(target, HUE)) \quad (7)$$

$$d_S(i) = diff(H(i, SAT), H(target, SAT)) \quad (8)$$

$$d_V(i) = diff(H(i, VAL), H(target, VAL)) \quad (9)$$

$$fitness_{HSV}(i) = 1 - 128\frac{d_H(i) + d_S(i) + d_V(i)}{3} \quad (10)$$

$$fitness_{AVERAGE}(i) = \frac{fitness_{RGB} + fitness_{HSV}}{2} \quad (11)$$

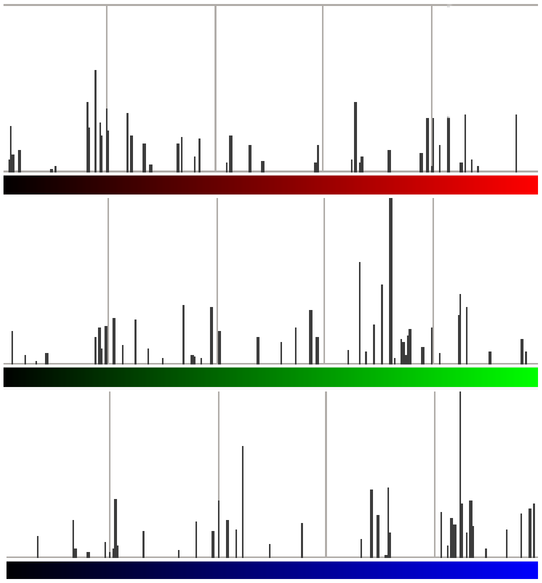


Fig. 4. RGB histogram of a solution generated by the algorithm.

D. Parameters used

A steady-state evolutionary algorithm has been used. Each individual is randomly generated at the initialization of the EA. The genome size is 50 elements (circles of maximum radium of 128 pixels). Population size has been set to 32 individuals. Uniform crossover rate is 0.5, and a binary tournament has been selected for selection (that is, a pool of 16 parents is selected and crossed). Mutation probability is 0.04 (the usual value of *1/genomesize*). Finally, the image size for each individual is 256x256 pixels. The individuals have been compared with the histograms obtained from the image of Figure 3 to guide the evolution.

V. Results

Each algorithm has been executed 30 times for each different fitness. Table II show the differences (and standard deviation) attained with each fitness used. As can be seen, using the HSV histogram differences as fitness produces a higher RGB similarity (and therefore, average) than using the RGB or Average fitness. However, using the average between the two histogram differences produces higher similiraty in HSV (0.294) that only taking into account the HSV. This can be explained because the HSV has QUE TIENE EL HSV QUE MOLE MAS. The maximum fitness is around 25% of similarity with the original image because, being a list of 50 circles, only a maximum of 50 different colours are used (while in the original jpg image can be more than millions). See the histogram of a generated best individual by the algorithm in Figure 4. An example of evolution for each fitness can be see in Figure 5, 6 and 7. Comparing with the RGB histogram as fitness, a more fluctuation in the HSV is produced (Figure 5). Although there are the same information in both histograms, the transformation from one to another is not linear, so there is no relation with the histograms of individuals generated during the evolution.

The best individuals attained are shown in Figure 8. Note than, although the fitness is similar, they produce different

TABLE II. RESULTS FOR THE DIFFERENT FITNESS. ONLY ONE HISTOGRAM TYPE IS USED, BUT THE OTHER VALUES OBTAINED ARE ALSO ADDED.

Differences used	Obtained RGB	Obtained HSV	Obtained Average
RGB Histogram	*0.267 ± 0.012*	0.170 ± 0.010	0.218 ± 0.009
HSV Histogram	0.227 ± 0.017	*0.265 ± 0.021*	0.246 ± 0.010
Average Histogram	0.173 ± 0.012	0.294 ± 0.013	*0.234 ± 0.010*

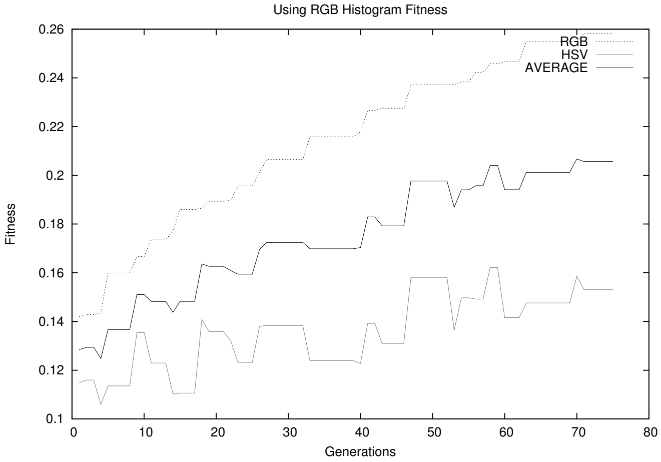


Fig. 5. Evolution of the difference in RGB histogram of the best individual compared with the test image.

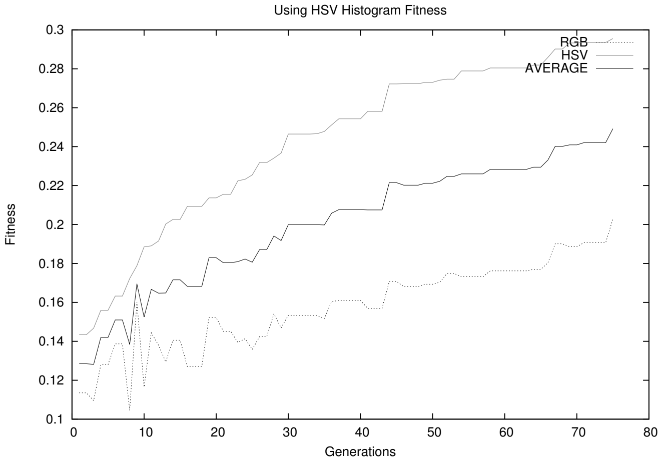


Fig. 6. Evolution of the difference in HSV histogram of the best individual compared with the test image.

color tones. This can be explained for the limitation of colors used in the individual. Figure 9 shows one evolution of the best individual using the HSV fitness in the first 64 generations.

VI. CONCLUSIONS AND FUTURE WORK

This paper introduces an Evolutionary Algorithm that uses the Processing framework to generate images and to extract image information. The advantages of this framework have been presented and we think that Evolutionary Art area can take advantage of them. In this work individuals are represented as a list of Processing primitives (circles) and the fitness functions used are based in the similarity with an existent aesthetic image (the comparison is performed using the libraries available in Processing). Three different fitness function using color histogram have been tested: difference with the HSV and RGB histograms, and an average difference of the two histograms at the same time. Experiments show that better results in terms of similarity are obtained using the HSV comparison. This is a basic image metric, only used by purposes of proof-of-concept and more complex measurements will be studied in next works.

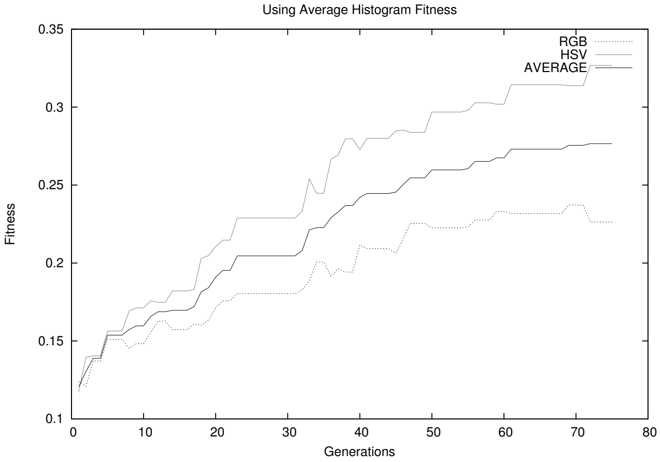


Fig. 7. Evolution of the difference of average of RGB and HSV histogram of the best individual compared with the test image.

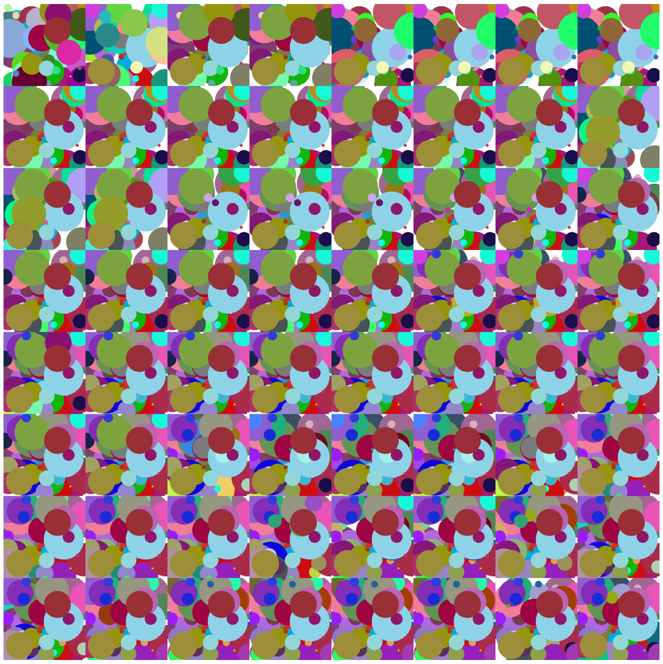
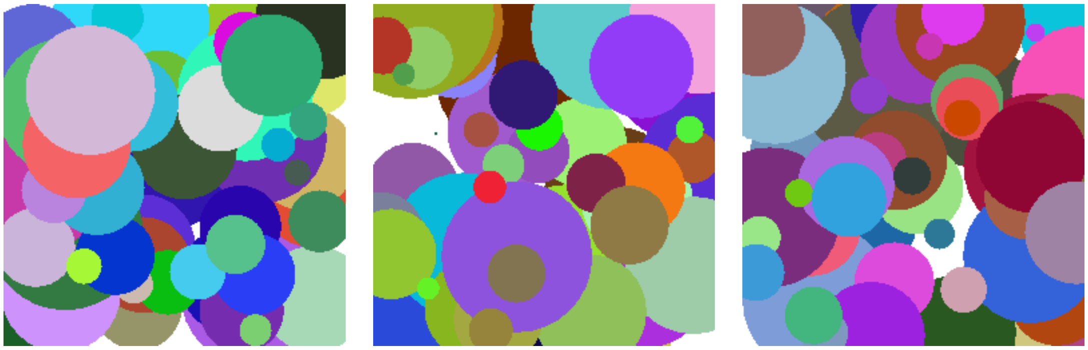


Fig. 9. Evolution of the best individual using the HSV histogram difference.

The future work for this research also includes more experiment with other kind of individuals, apart from circles: using other primitives, such as rectangles or triangles, for example. The use of textures and gradients will generate images with higher number of colors, obtaining more fidelity (more than



(a) Best individual using RGB

(b) Best individual using HSV

(c) Best individual using AVERAGE

Fig. 8. Best individuals obtained with the three fitness used (HSV, RGB and AVERAGE).

25%) with the test image. Other metrics explained in previous sections will be also implemented. Finally, our intention is not create only static images, but use the Processing libraries to create evolutionary interactive art combining sounds and motion. A human guidance tool is also being developed to obtain human feedback to create a knowledge base for future experimentation (available in http://evorq.ugr.es:8080/HumanGuidance).

The used software and algorithms presented are Open Source under a GPL license, and can be obtained from http://www.osgiliath.org.

ACKNOWLEDGEMENTS

This work has been supported in part by FPU research grant AP2009-2942 and projects EvOrq (TIC-3903), SINECA (0100DGT21285, Spanish Direccion General de Trafico), TIN2011-28627-C04-02 and FFI2011-22397 (funded by Spanish Ministry for Economy and Productivity). This work was developed during the 5 Hackathon of the Office of Open Software of the University of Granada (http://sl.ugr.es/hackathon), where several members of different disciplines collaborated during their creation.

REFERENCES

[1] D. W. Corne and P. J. Bentley, *Creative evolutionary systems*. Morgan Kaufmann, 2001.

[2] A. Eiben and J. Smith, "What is an evolutionary algorithm?" in *Introduction to Evolutionary Computing*, G. Rozenberg, Ed. Addison Wesley, 2005, pp. 15–35.

[3] F. Hoenig, "Defining computational aesthetics," in *Proceedings of the First Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, ser. Computational Aesthetics'05. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2005, pp. 13–18. [Online]. Available: http://dx.doi.org/10.2312/COMPAESTH/COMPAESTH05/013-018

[4] S. DiPaola and L. Gabora, "Incorporating characteristics of human creativity into an evolutionary art algorithm," *Genetic Programming and Evolvable Machines*, vol. 10, no. 2, pp. 97–110, 2009.

[5] P. Galanter, "Computational aesthetic evaluation: past and future," in *Computers and Creativity*. Springer, 2012, pp. 255–293.

[6] G. Birkhoff, *Aesthetic Measure 1933*. Harvard University Press, 2003. [Online]. Available: http://books.google.es/books?id=F8A-UF8QgckC

[7] E. Den Heijer and A. Eiben, "Comparing aesthetic measures for evolutionary art," *Applications of Evolutionary Computation*, pp. 311–320, 2010.

[8] Y. Li, C. Hu, M. Chen, and J. Hu, "Investigating aesthetic features to model human preference in evolutionary art," *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pp. 153–164, 2012.

[9] E. Del Acebo and M. Sbert, "Benford's law for natural and synthetic images," in *Computational Aesthetics in Graphics, Visualization and Imaging*. The Eurographics Association, 2005, pp. 169–176.

[10] E. den Heijer and A. Eiben, "Evolving pop art using scalable vector graphics," *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pp. 48–59, 2012.

[11] S. Draves, "The electric sheep," *ACM SIGEVOlution*, vol. 1, no. 2, pp. 10–16, 2006.

[12] C. Aguilar and H. Lipson, "A robotic system for interpreting images into painted artwork," in *Proceedings of the 11th Generative Art Conference (GA2008), Politecnico di Milano University, Milan, Italy*, 2008.

[13] K. Sims, "Evolving virtual creatures," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 1994, pp. 15–22.

[14] P. Machado and A. Cardoso, "Computing aesthetics," *Advances in Artificial Intelligence*, pp. 105–119, 1998.

[15] C. Reas and B. Fry, *Processing: A Programming Handbook for Visual Designers and Artists*, ser. Working paper series (National Bureau of Economic Research). Mit Press, 2007. [Online]. Available: http://books.google.es/books?id=tqW75bfJkxIC

[16] P. García-Sánchez, J. González, P. Castillo, M. Arenas, and J. Merelo-Guervós, "Service oriented evolutionary algorithms," *Soft Computing*, pp. 1–17, 2013, 10.1007/s00500-013-0999-5. [Online]. Available: http://dx.doi.org/10.1007/s00500-013-0999-5