Evolutionary Art using Processing

Author 1
Affiliation 1
University of Granada
Email:

Author 2
Affiliation 2
University of Granada
Email:

Author 3
Affiliation 3
University of Granada
Email:

***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 pressence 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 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 latter increases the similirities of the RGB and HSV with respecting to using them separately.**

I. INTRODUCTION

Evolutionary art blah, blah, blah, ...

The main goal of this paper is ... We show ...

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

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" [1]. In the field of computational aesthetics, evolutionary systems can play an important role, enabling the evolution of aesthetically pleasing or innovative structures [2].

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 [3]:

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 [3], 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 [4].
  - 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 fount in Table I.

III. PROCESSING AND IMAGE METRICS

In this section we will describe Processing [1]. Processing [13] 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:

[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 [5], Image order [6], Benford Law [7]
Based in Design Principles	Color contrast (hue) [8], Color ingredient [6], Composition, tonality and color [2].
Interactive Evolutionary Computation	The electric sheep project [9]
Error relative to Exemplars	Resemblance score [2], pixel comparation [10]
Performance based goals	Evolving virtual creatures [11]
Complexity measures	Image complexity [6], Machado and Cardoso aesthetic measure [12]

- 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, and can work with the community in its development.
- It is based in OpenGL, obtaining a good 3D acceleration.
- Includes more than 100 libraries for video, sound, phisycs, 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 are: BLABLABLABLA

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.

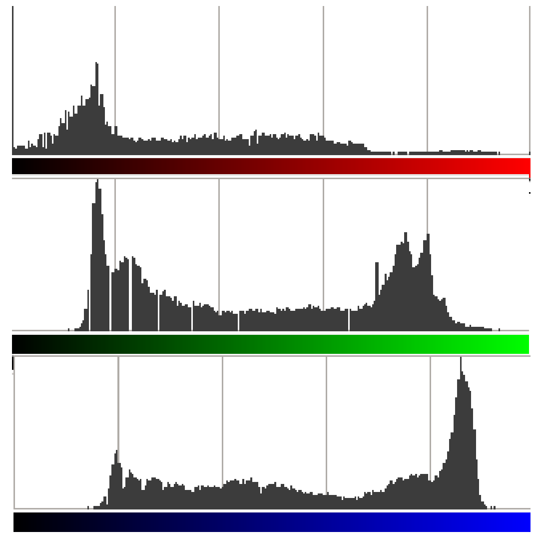


Fig. 2. RGB histogram of the Figure III.

- 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.

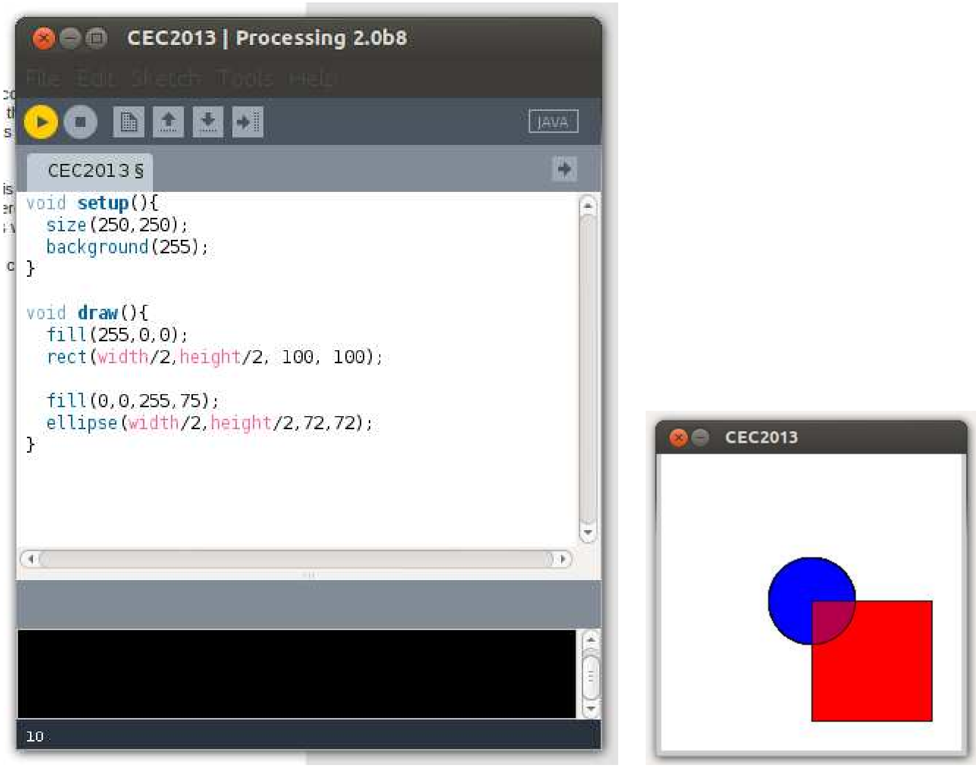
The Color module can be used to analyze images taking into account their histogram. The color histogram represents the frequency of occurrence of each color intensities present in the image, by accounting for such sharing pixels color intensity values.

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). The resolution of the various components is not uniform but used an increased number of bits for representing the hue component, which for the two remaining two bits being sufficient in the case of Value. Figure III shows the RGB histogram of the image in the Figure III.

IV. EXPERIMENTAL SETUP

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 **??** is accessed extending the *PApplet* class. In this work, Processing has been



(a) Processing IDE

(b) Runtime of the code.

Fig. 1. Processing IDE and sketch execution.

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

integrated to an existent EA framework, OSGiLiath [14], 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 publicly-available source code of OSGiLiath (available in http://www.osgiliath.org) under a GPL License.

B. Individual representation

To test the Processing advantages and perform the experiments, the individual is a list of Processing Circles.

C. Fitness used

For this piece of research, we focused on two measures of aesthetics: basic histogram comparison and image matching. The fitness functions are included in the "Error relative to Exemplars" category, using Galanter [3] classification.

An histogram is a graphical representation of the tonal distribution in an image. The histogram for the property i is computed following (**??**).

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

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

$$d_R(i) = diff(H(i, RED), H(target, RED)) \tag{3}$$

$$d_G(i) = diff(H(i, GREEN), H(target, GREEN)) \tag{4}$$

$$d_B(i) = diff(H(i, BLUE), H(target, BLUE)) \tag{5}$$

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

$$d_H(i) = diff(H(i, HUE), H(target, HUE)) \tag{7}$$

$$d_S(i) = diff(H(i, SAT), H(target, SAT)) \tag{8}$$

$$d_V(i) = diff(H(i, VAL), H(target, VAL)) \tag{9}$$

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

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

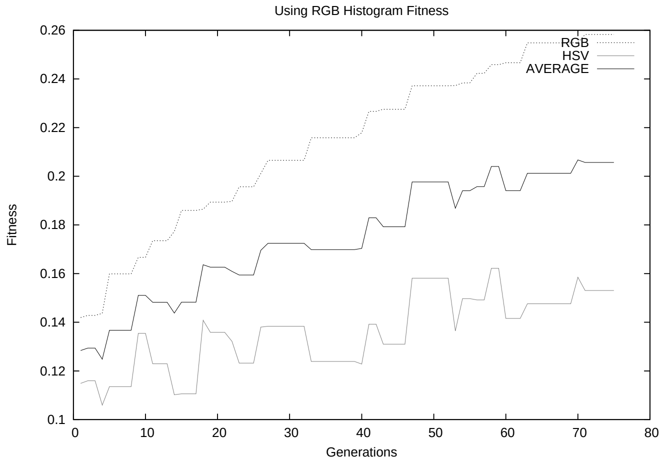


Fig. 4. Evolution of the difference in RGB histogram of the best individual compared with the test image.

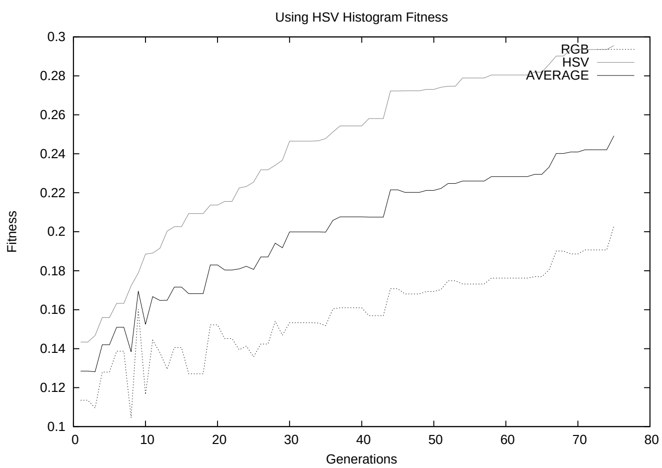


Fig. 5. Evolution of the difference in HSV histogram of the best individual compared with the test image.

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. Crossover rate is 0.5, and a binary tournament has been selected for selection (that is, a pool of 16 parents is selected). Mutation probability is 0.04 (1/genomesize). Finally, the image size for eac individual is 256x256 pixels. The invidivuals have been compared with the histograms obtained from the image of Figure III.

V. Results

Table II show the differences attained with each fitness used. As can be see... An example of evolution for each fitness can be see in Figure V, V and V.

The best individuals attained are shown in Figure V.

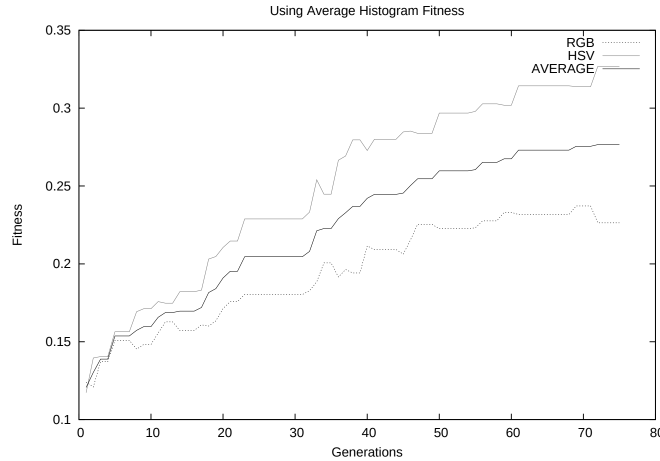


Fig. 6. Evolution of the difference of average of RGB and HSV histogram of the best individual compared with the test image.

VI. Conclusions and Future Work

This paper introduces an Evolutionary Algorithm that uses Processing to generate images and to extract image information. Individuals are represented as a list of Processing primitives and the fitness functions used are based in the similarity with an existent aesthetic image. Three different fitness function using histogram have been tested: difference with the HSV and RGB histograms, and an average difference of the two histograms at the same time.

The future work for this research work includes more experiment with other kind of individuals, appart from circles: using other primitives, such as rectangles or triangles, for example. With the use of textures and gradients would include a larger histogram EXTENDIDO. 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.

The used software is Open Source and can be obtained in 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) and TIN2011-28627-C04-02.

References

[1] 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

[2] 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.

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

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*

(a) Best individual using RGB (b) Best individual using HSV (c) Best individual using AVERAGE

Fig. 7. Best individuals obtained with the three fitness used (HSV, RGB and AVERAGE).

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

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

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

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

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

[13] 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

[14] 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