# Expression-Based Evolution of Faces

Penousal Machado[1], João Correia[1], and Juan Romero[2]

[1] CISUC, Department of Informatics Engineering, University of Coimbra, 3030 Coimbra, Portugal machado@dei.uc.pt, jncor@dei.uc.pt
[2] Faculty of Computer Science, University of A Coruña, Coruña, Spain jj@udc.pt

**Abstract.** The combination of a classifier system with an evolutionary image generation engine is explored. The framework is instantiated using an off-the-shelf face detection system and a general purpose, expression-based, genetic programming engine. By default, the classifier returns a binary output, which is inadequate to guide evolution. By retrieving information provided by intermediate results of the classification task, it became possible to develop a suitable fitness function. The experimental results show the ability of the system to evolve images that are classified as faces. A subjective analysis also reveals the unexpected nature and artistic potential of the evolved images.

**Keywords:** Evolutionary Art, Automatic Fitness Assignment, Face Detection

## 1 Introduction

In theory most expression based, Sims like [16], Evolutionary Art (EA) systems have the ability to generate any image of any kind [9, 11]. In practice, the images they tend to evolve depend on the used representation scheme. For this reason, the production of expression-based EA systems tends to be dominated by abstract images. In our view this does not constitute a problem. Nevertheless, the desire to evolve figurative images by evolutionary means is present since the early years of EA, e.g. Steven Rooke [21], and has not faded (see, e.g., [7]).

The issue has been tackled by two main types of approach: (i) Developing tailor EA systems which resort to representations that promote the discovery of figurative images, usually of a certain kind; (ii) Use general purpose EA systems and develop fitness assignment schemes that guide the system towards figurative images. We are particularly interested in the second approach.

In the past few years, object detection systems, in particular face detection, have become a hot topic of interest and research. Applications that employ this kind of systems are becoming widespread. For instance, they can be found in search engines, social networks, incorporated in cameras, or in applications for smart phones.

Romero et al. [14] suggests combining a general purpose evolutionary art system with an image classifier trained to recognize faces, or other types of objects, to evolve images of human faces. Nowadays availability of off-the-shelf face detectors makes that approach possible. As such, the goal of the present research is the evolution of images of human faces by means of a general-purpose, expression-based, EA system.

Our approach is informed by previous research, e.g. [2, 15, 10], where classifier systems, namely Neural Networks, are used to guide the evolutionary runs. However, among others, our approach possesses the following discriminating characteristics:

- Using an off-the-shelf classifier instead of one developed purpose-built to guide evolution;
- The goal is to evolve specific figurative images, i.e. faces, while the mentioned classifiers try to assess aesthetics, style or novelty;
- A Haar Cascade classifier (see Viola et al. [20]) is used instead of Neural Networks;

The paper is structured as follows: In the following section we make a brief overview of related work; In section 3 we describe our approach for the evolution of face images describing the overall framework, the Genetic Programming (GP) engine, the face detection system, and fitness assignment, section 3.3; Section 4 presents the experimental setup, the results attained and their analysis; Finally, in section 5 we draw overall conclusions and indicate future research.

## 2 Related Work

The use of Evolutionary Computation (EC) for the evolution of human faces is not new. Caldwell and Johnston [6] used a Genetic Algorithm (GA) to recombine portions of existing face images, in an attempt to build a criminal sketch artist. With similar goals, Frowd and Hancock [5] use a GA, Principal Components Analysis and eigenfaces to evolve human faces. In contrast with this approaches that attempt to create photographic human face images, Baker [1] focuses on the evolution of line drawings, using a GP approach. The evolution of cartoon faces [12] and cartoon face animations [7] through GAs has also been explored. Additionally, Lewis also explored the evolution of human figures.

All the previously mentioned approaches share a common aspect, the system has been specifically designed for the evolution of human faces. The work of Baker is an exception, the system can evolve other types of line drawings, however the system was initialized with hand-built line drawings of human faces.

This approach contrasts with the ones used by Ventrella [19] and DiPaola [3] where a general purpose evolutionary art tool is used. Both approaches are akin to a classical symbolic regression problem in the sense that a target image exists and the similarity between the evolved images and the target image is used to assign fitness. In addition to similarity, DiPaola also considers expressiveness when assigning fitness. This approach results in images with artistic potential, which was the primary goal of these approaches, but that would hardly be classified as human faces. As far as we know, the difficulty to evolve a specific target image, using symbolic regression inspired approaches, is common to all “classical” expression-based GP systems.

Our research is also informed by previous works where a classifier system is used to assign fitness, namely: by the seminal work of Baluja et al. [2], where a Neural Network trained to replicate the aesthetic assessments is used; by the work of Saunders and Gero [15], which employs a Kohonen Self-Organizing network to determine novelty; by the work of Machado et al. [10] where a bootstrapping approach, relying on a neural network, is used to promote style changes among evolutionary runs.

## 3 Overview of the Approach

Figure 1 presents an overview of the framework, which is composed of two main modules, an evolutionary engine and a classifier.

Execution proceeds as follows:

1. Random initialization of the population;
2. Rendering of the individuals, i.e., genotype-phenotype mapping;
3. Apply the classifier to each phenotype;
4. Use the results of the classification to assign fitness; This may, and in our case does, require assessing internal values and intermediate results of the classification;
5. Select progenitors; Apply genetic operators, create descendants; Use the replacement operator to update the current population;
6. Repeat from 2 until some stopping criterion is met.

For the purposes of this paper the framework was instantiated with a general-purpose GP-based image generation engine – described in section 3.1 – and with a Haar Cascade Face Detector – described in section 3.2. To create a fitness function able to guide evolution it is necessary to convert the binary output of the face detector to one that can provide suitable fitness landscape. This is attained by accessing internal results of the classification task that give an indication of the degree of certainty in the classification. This process is described in section 3.3.

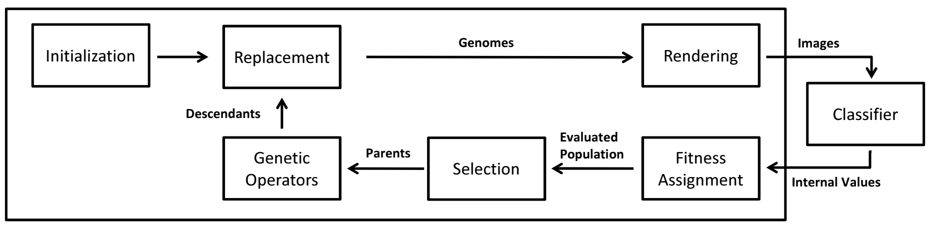


Fig. 1: Overview of the system.

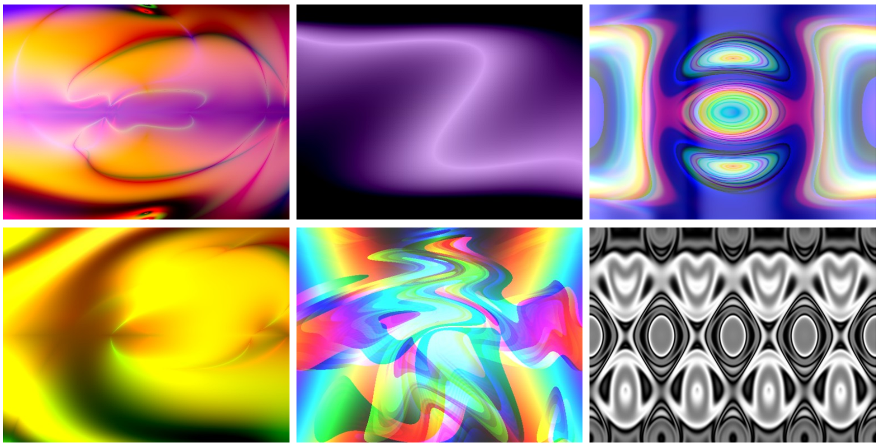

Fig. 2: Examples of images generated by the evolutionary engine using interactive evolution.

### 3.1 Genetic Programming Engine

The EC engine used in these experiments is inspired by the works of Sims [16]. It is a general purpose, expression-based, GP image generation engine that allows the evolution of populations of images. The genotypes are trees composed from a lexicon of functions and terminals. The function set is composed of simple functions such as arithmetic, trigonometric and logic operations. The terminal set is composed of two variables, $x$ and $y$, and random constant values. The phenotypes are images that are rendered by evaluating the expression-trees for different values of $x$ and $y$, which serve both as terminal values and image coordinates. In other words, to determine the value of the pixel in the (0,0) coordinates one assigns zero to $x$ and $y$ and evaluates the expression-tree. A thorough description of the GP engine can be found in [10].

Figure 2 displays typical imagery produced via interactive evolution using this EC system.

### 3.2 Face Detection

For classification purposes we use Haar Cascade classifiers (see Viola et al. [20]) built to detect frontal faces. The code and executables are included in the OpenCV API[3]. This classification approach was chosen due to its state of the art relevance and for its fast classification. This algorithm uses a set of small features in combination with a variant of the Adaboost [4], and is able to attain efficient classifiers. The classifiers assume the form of a cascade of small and simple classifiers that use Haar features [13], i.e., rectangular features, that are calculated through the integral image method.

[3] OpenCV — `http://opencv.willowgarage.com/wiki/`

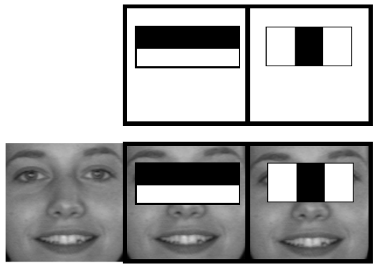

Fig. 3: Haar features (adapted from [20]).

An example of the feature identification process is presented in figure 3. In this case, two-rectangular features were used. By subtracting the pixels of the black zone with the white zone we obtain the feature result. If this result is superior to a given threshold, then the tested feature is present on the image. Initially, only vertical and horizontal features were used, the work of Lienhart [8] introduced several extensions to the used features, including oblique features.

The face detection process can be summarized in these steps (parameters from Viola et al. work [20]):

1. Define a window of size $w$ ($20 \times 20$).
2. Define a scale factor $s$ greater than 1. For instance 1.2 means that the window will be enlarged by 20%.
3. Define $W$ and $H$ has the size of the input image.
4. From $(0, 0)$ to $(W, H)$ define a sub-window with a starting size of $w$ for calculation.
5. For each of these sub-windows apply the cascade classifier. The cascade has a group of stage classifiers, as represented in figure 4. Each stage is composed, at its lower level, of a group of Haar features. Apply each feature of each stage to the sub-window. If the resulting value is lower than the stage threshold the sub-window does not have a face and the search terminates for the sub-window. If it is higher continue to next stage. If all stages are passed, the sub-window has a face.
6. Apply the scale factor $s$ to the window size $w$ and repeat 5 until window size exceeds the image in at least one dimension.

### 3.3 Fitness Assignment

The process of fitness assignment is crucial from an evolutionary point of view, and therefore it holds a large importance for the success of the described system. The goal is to evolve images that the face detector classifies as faces. However, the face detector returns a binary output, which is inappropriate to guide evolution. A binary function gives no information of how close an individual is to being a valid solution to the problem and, as such, with a binary function the EA would

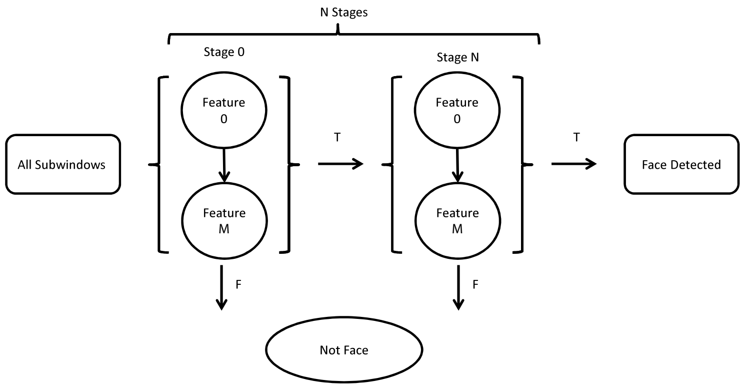


Fig. 4: Cascade of classifiers with N stages, adapted from [8].

be performing a random search. It is, therefore, necessary to extract additional information from the face detection process in order to build a suitable fitness function.

This is attained by accessing internal results of the classification task that give an indication of the degree of certainty in the classification. In several informal experiments, we focused on developing an appropriate fitness function by analyzing the results of several runs, by trial and error, by incremental improvements and refinements, etc. We eventually settled for the following formula, which takes advantage of the cascade structure of the classifier:

$$fitness(x) = \sum_{i}^{countstages_x} beststagedifference_x(i) * i + countstages_x * 10 \quad (1)$$

In a nutshell, images that go through several classification stages, and that hence may be close to be classified as a face, have higher fitness than those rejected in early stages. Variables $countstages_x$ and $beststagedifference_x(i)$ are extracted from the face detection algorithm. Variable $countstages_x$, holds the number of stages that image, $x$, has successfully passed in the cascade of classifiers. The rationale is the following, an image that passes several stages is likely to be closer of being recognized as having a face than one that passes fewer stages. In other words, passing several stages is a pre-condition to being identified as a face image. Variable $beststagedifference_x(i)$ holds the maximum difference between the threshold necessary to overcome stage $i$ and the value attained by the image at the $i^{th}$ stage. Images that are clearly above the thresholds are preferred over ones that are only slightly above them. Obviously, this fitness function is only one of the several possible ones.

Table 1: Parameters of the GP engine. See [10] for a detailed description

| Parameter | Setting |
|---|---|
| Population size | 50 |
| Number of generations | 100 |
| Crossover probability | 0.8 (per individual) |
| Mutation probability | 0.05 (per node) |
| Mutation operators | sub-tree swap, sub-tree replacement, node insertion, node deletion, node mutation |
| Initialization method | ramped half-and-half |
| Initial maximum depth | 5 |
| Mutation max tree depth | 3 |
| Function set | $+$, $-$, $\times$ , /, min, max, abs, neg, warp, sign, sqrt, pow, mdist, sin, cos, if |
| Terminal set | $x$, $y$, random constants |

## 4 Experimentation

In order to conduct the experiments described in this paper, three classifiers were used. These were obtained from Lienhart's [8] website[4] and will be named C1, which uses the "alt.xml" file; C2 ("alt2.xml"); C3('default.xml"). We performed 30 independent evolutionary runs for each of these classifiers. The settings of the GP engine, presented in table 1, are similar to those used in previous experimentation in different problem domains. Since the used classifiers only deal with greyscale information, the GP engine was also limited to the generation of greyscale images for the scope of these experiments.

Figure 5 summarizes the results attained in terms of mean fitness and maximum fitness per run. Since the fitness values attained by different classifiers are not comparable, the values are normalized by dividing the raw fitness by the mean's maximum achieved in each classifier's test. Each chart displays the fitness attained by the classifier used to guide fitness and also the fitness that would be assigned by the two classifiers that had no interference in the run.

An analysis of these charts reveals interesting aspects concerning similarity among classifiers. As it can be observed, the curves of classifiers C1 and C2 vary in similar ways, particularly in terms of maximum average fitness, independently of which classifier is guiding the run, which indicates that these classifiers are strongly correlated. In contrast, fitness according to classifier C3 only reaches high values when C3 is used to guide the evolutionary runs. As a whole these results suggest that classifier C3 is more robust than C1 and C2, in the sense that it is less likely to classify non-face images as faces. Viola and Jones [20] arrive to a similar conclusion based on experiments done in a non evolutionary context.

The GP engine was able to find images classified as faces in all of the 90 performed runs. However, and somewhat surprisingly, from a human perspec-

[4] Haar Cascades [8] – `http://alereimondo.no-ip.org/OpenCV/34`

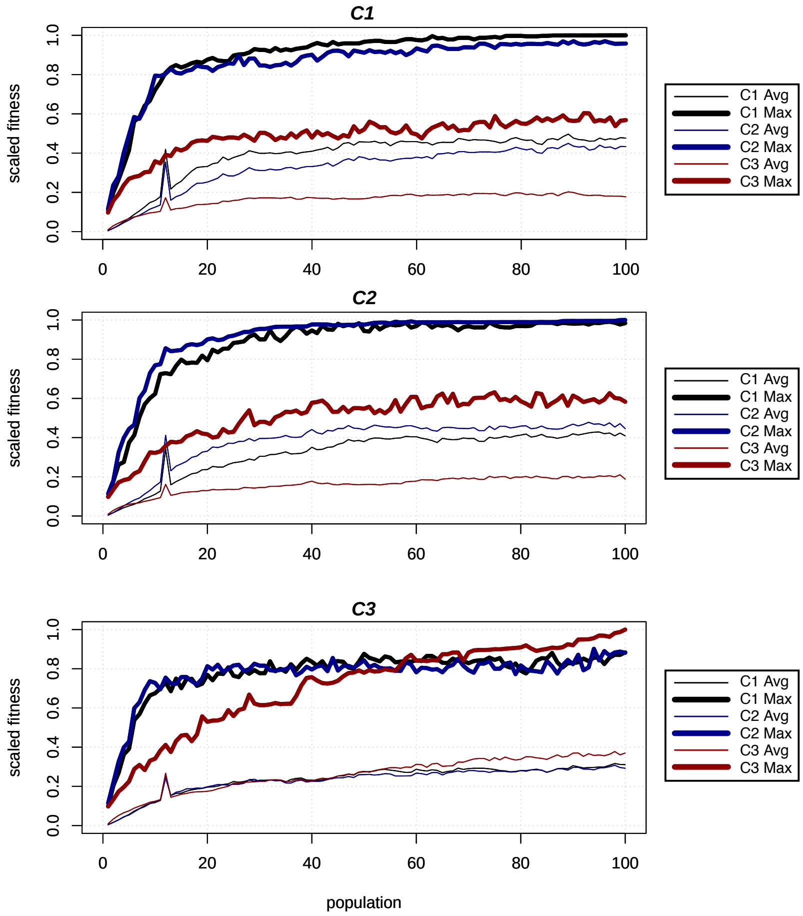


Fig. 5: Evolution of the average and maximum fitness when using C1 (top) C2 (middle) and C3 (bottom) to assign fitness. Results are averages of 30 independent runs.

tive most of the runs did not evolve images that look like faces (obviously this statement has a degree of subjectivity). Thus, in most evolutionary runs the GP engine exploited the limitations of the classifier and found "shortcuts" that allowed it to improve fitness, and evolve images that are classified as faces, without evolving images that actually look like faces (see figure 6). The ability of EC to find such shortcuts and exploit weaknesses of the fitness assignment scheme has been reported on previous studies (see, e.g., [17, 18, 10]). These results open a series of possibilities, including the use of this approach to assess the robustness of face detection systems, and also the use of evolved images as part of the training set of these classifiers in order to remedy some of their shortcomings.

Fig. 6: Examples of evolved images identified as faces by the classifiers that do not resemble faces from a human perspective.

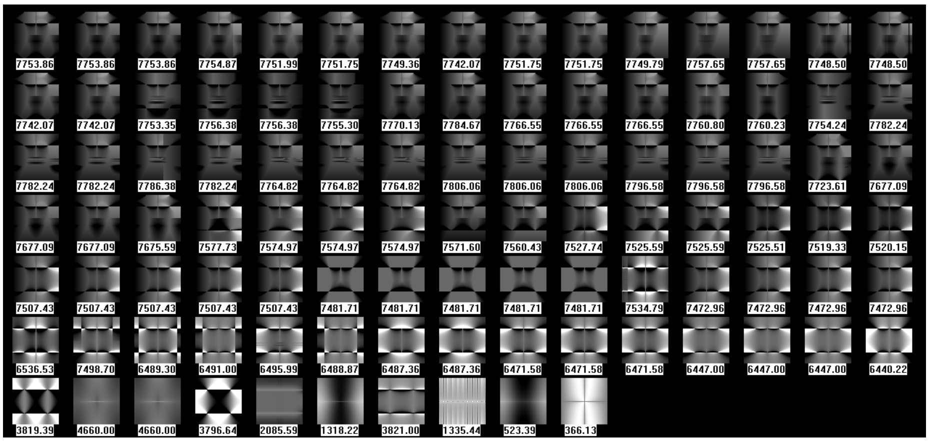


Fig. 7: Fittest individual per generation. From the last to the first generation in reading order.

Although we are pursuing that line of research, it is beyond the scope of the current paper.

According to our subjective assessment, some of the runs (5 using C1, 4 using C2 and 5 using C3) were able to find images that resemble a frontal human face. Figure 7 shows the evolution of the best individual, per generation, during the course one of those runs.

In figure 8 we show some of the most interesting evolved images (according to the authors). These results show the ability of the GP engine to create figurative images, which are reminiscent of human faces. Several of these images are evocative of faces of cartoon characters (e.g. the second image of the first row of figure 8, which has been described by several of our co-workers as Wolverine's face) and african masks (e.g., the last image from the first row figure 8). This result may reveal a tendency towards the exaggeration of facial features, and hence caricature, which is consistent with the fitness assignment scheme, in the sense that marked facial features may promote face detection.

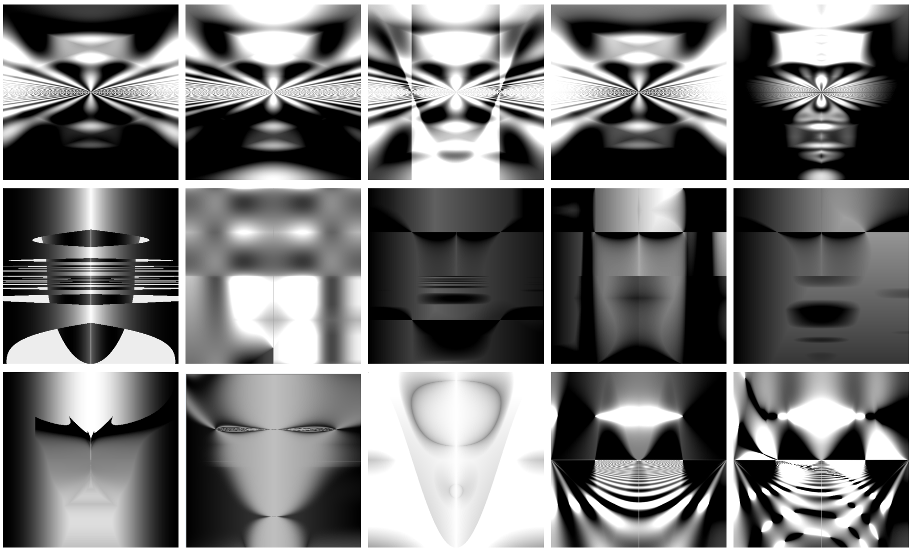

Fig. 8: Examples of some of the most interesting images that have been evolved.

## 5 Conclusions

The goal of the current paper was to generate figurative images by evolutionary means, without resorting to representations specifically tailored to promote the evolution of images of a certain kind. A framework for the evolution of figurative images is presented and explored, using a general-purpose expression-based GP image generation engine and off-the-shelf face detector systems. Internal results of the classification task are employed to build a fitness function.

The experimental results attained in 90 independent evolutionary runs show the ability of the GP engine to find and exploit shortcomings of the classifier systems. They also demonstrate the ability of the framework to evolve images that are evocative of human faces and masks.

Although the classifiers used in these experiments are face detectors, Haar Cascade classifiers [20] can, and have been used to detect other objects. Therefore, using the same framework with different classifiers, it should be possible to evolve figurative images evocative of other types of object. We are currently conducting such experiments.

The images evolved in different runs can be combined, refined and explored for artistic purposes by using user-guided evolution or automatic fitness assignment schemes, which take into account aesthetic or stylistic properties. In this regard, the plasticity of the expression-based representation may be a valuable asset.

Finally, the ability of EC to find shortcomings of the classifier can be used to assess and improve classifier performance. This can be achieved by evolving images that are incorrectly classified and then using these images as part of the negative training set of the classifier. This line of research was triggered by the findings described herein. The experimental results attained are promising showing relevant increases of performance.

## 6 Acknowledgments

This research is partially funded by: the Portuguese Foundation for Science and Technology, project PTDC/EIA–EIA/115667/2009; the Spanish Ministry for Science and Technology, project TIN2008–06562/TIN; Xunta de Galicia, project XUGA–PGIDIT10TIC105008PR.

## References

1. E. Baker. Evolving line drawings. Technical Report TR-21-93, Harvard University Center for Research in Computing Technology, 1993.
2. S. Baluja, D. Pomerlau, and J. Todd. Towards automated artificial evolution for computer-generated images. *Connection Science*, 6(2):325–354, 1994.
3. S. R. DiPaola and L. Gabora. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines*, 10(2):97–110, 2009.
4. Y. Freund and R. E. Schapire. A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting, 1995.
5. C. Frowd and P. Hancock. Evolving human faces. In J. Romero and P. Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 189–210. Springer Berlin Heidelberg, 2007.
6. V. S. Johnston and C. Caldwell. Tracking a criminal suspect through face space with a genetic algorithm. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages G8.3:1–8. Institute of Physics Publishing and Oxford University Press, Bristol, New York, 1997.
7. M. Lewis. Evolutionary visual art and design. In J. Romero and P. Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 3–37. Springer Berlin Heidelberg, 2007.
8. R. Lienhart and J. Maydt. An Extended Set of Haar-Like Features for Rapid Object Detection. In *IEEE ICIP 2002*, pages 900–903, 2002.
9. P. Machado and A. Cardoso. All the truth about NEvAr. *Applied Intelligence, Special Issue on Creative Systems*, 16(2):101–119, 2002.
10. P. Machado, J. Romero, and B. Manaris. Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In J. Romero and P. Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 381–415. Springer Berlin Heidelberg, 2007.
11. J. McCormack. Facing the future: Evolutionary possibilities for human-machine creativity. In J. Romero and P. Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 417–451. Springer Berlin Heidelberg, 2007.
12. K. Nishio et al. Fuzzy fitness assignment in an interactive genetic algorithm for a cartoon face search. In E. Sanchez, T. Shibata, and L. A. Zadeh, editors, *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives*, volume 7. World Scientific, 1997.

13. C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision*, pages 555–562, Jan. 1998.
14. J. Romero, P. Machado, A. Santos, and A. Cardoso. On the development of critics in evolutionary computation artists. In R. Günther et al., editors, *Applications of Evolutionary Computing, EvoWorkshops 2003: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 2611 of *LNCS*, Essex, UK, 2003. Springer.
15. R. Saunders and J. Gero. The digital clockwork muse: A computational model of aesthetic evolution. In G. Wiggins, editor, *AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science*, pages 12–21, York, UK, 2001.
16. K. Sims. Artificial evolution for computer graphics. *ACM Computer Graphics*, 25:319–328, 1991.
17. L. Spector and A. Alpern. Criticism, culture, and the automatic generation of artworks. In *Proceedings of Twelfth National Conference on Artificial Intelligence*, pages 3–8. AAAI Press/MIT Press, Seattle, Washington, USA, 1994.
18. A. Teller and M. Veloso. Algorithm evolution for face recognition: what makes a picture difficult. In *Evolutionary Computation, 1995., IEEE International Conference on*, 1995.
19. J. Ventrella. Self portraits with mandelbrot genetics. In *Proceedings of the 10th international conference on Smart graphics*, SG'10, pages 273–276, Berlin, Heidelberg, 2010. Springer-Verlag.
20. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511, 2001.
21. L. World. Aesthetic selection: The evolutionary art of steven Rooke. *IEEE Computer Graphics and Applications*, 16(1), 1996.