

Semantic Aware Methods for Evolutionary Art

Penousal Machado
CISUC, Department of Informatics Engineering
University of Coimbra, 3030
Coimbra, Portugal
machado@dei.uc.pt

João Correia
CISUC, Department of Informatics Engineering
University of Coimbra, 3030
Coimbra, Portugal
jncor@dei.uc.pt

ABSTRACT

In the past few years the use of semantic aware crossover and mutation has become a hot topic of research within the Genetic Programming community. Unlike traditional genetic operators that perform syntactic manipulations of programs regardless of their behavior, semantic driven operators promote direct search on the underlying behavioral space. Based on previous work on semantic Genetic Programming and Genetic Morphing, we propose and implement semantic driven crossover and mutation operators for evolutionary art. The experimental results focus on assessing how these operators compare with traditional ones.

Categories and Subject Descriptors

I.3.3 [Computer Graphics]: Picture/Image Generation;
I.2.m [Artificial Intelligence]: Miscellaneous

Keywords

Semantic Operators, Evolutionary Art, Genetic Programming, Computational Creativity

1. INTRODUCTION

The use of geometric semantic genetic operators [14] in the context of Genetic Programming (GP) is a recent field of research. While conventional GP perform syntactic changes to programs ignoring their behaviour, geometric semantic operators consider behaviour, thus allowing the direct search of the behavioral space. The theoretical advantages of such operators are huge: they induce a unimodal fitness landscape for all the problems consisting in matching input data with known outputs.

We revisit the early works of Sims [18] and Hart [7] concerning the generation of short animations through “genetic morphing” between individuals. We adapt their approaches, developing several semantic-inspired crossover and mutation operators. We use these operators in the context of evolutionary art and study their properties.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO’14, July 12–16, 2014, Vancouver, BC, Canada.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2662-9/14/07 ...\$15.00.

<http://dx.doi.org/10.1145/2576768.2598293>.

In Section 2 we make a short overview of geometric semantic operators proposed in GP literature, introducing several important concepts. Section 3 makes a short introduction to expression based evolutionary art and an overview of our evolutionary art system. The semantic-inspired genetic operators proposed in this paper are described in Section 4 and the results obtained through their use are summarized on Section 5. Finally we draw conclusions.

2. GEOMETRIC SEMANTIC OPERATORS

Traditional GP crossover and mutation operators perform syntactic manipulations of programs – e.g. sub-tree swapping – ignoring the semantics of the programs that they manipulate [14]. While semantic-blind operators have shown their merit, as it is demonstrated by the extensive list of valuable results and breakthroughs obtained by GP approaches, the definition of operators that are aware of the semantics of candidate solutions [2] is becoming an important area of research within the GP community.

The exact meaning of semantics in the context of GP is debatable, with several researchers adopting the definition proposed by Uy et al. [20] and Moraglio et al. [14] who consider semantics as the function computed by the program, more precisely, as the behavior of a program over a set of data, i.e., “input-output pairs making up the computed function”. In simple terms, adopting this definition, where semantics equates behavior and applying it to expression-based evolutionary art, implies that the semantics/behavior of an individual is the image produced by the individual when executed over a set of x, y coordinate values.

Since traditional GP crossover and mutation ignore semantics, the consequences of the syntactic modifications performed by these operators on the behavior of the programs are difficult to predict. Moraglio et al. [14] recently introduced a new type of genetic operators, called geometric semantic genetic operators, that manipulate directly the behavior of the programs, ignoring their syntactic information. The ability to directly manipulate semantic information confers them several interesting properties, most notably: they induce a unimodal fitness landscape for all the problems consisting in matching input data with known target outputs. In other words, under these operators problems such as symbolic regression become trivial, in the sense that a program that reproduces the training data is found with little effort.

Unfortunately, they also possess a major disadvantage: these operators always produce offspring that are larger than their parents, which causes an exponential growth in program size. Thus, while it is trivial to solve problems such

as symbolic regression, the size of the evolved programs is so large that, for most applications, it defeats the purpose. Although it is common to define symbolic regression as the problem of finding a mathematical expression that approximates or matches the training data, this definition is incomplete. Matching the training data is only a part of the problem and it can be solved trivially by a variety of methods, e.g. considering that your training data consists of n (x, y) pairs, with $x, y \in \mathbb{R}$ you can always use polynomial regression to derive polynomial of order $n - 1$ that fits the data perfectly. Thus, the goal of symbolic regression is better defined as the problem of finding a **compact** expression that **approximates** the training data and that **generalizes** well. Analogously, in the context of evolutionary art, as demonstrated by Machado and Cardoso [12] finding the symbolic expression for the Mona Lisa, or any other given image, is both a trivial problem and a fruitless endeavor *per se*. However, finding a compact representation is a difficult and valuable problem, perhaps even central for evolutionary art [13], opening the door for a wide range of applications.

Considering the potential and the limitations of semantic operators their contribution to the field of evolutionary art is unclear, and should be scrutinized. Next we describe the semantic geometric crossover and mutation operators for real functions presented by Moraglio et al. [14].

Geometric Semantic Crossover: Given two parent functions $T_1, T_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, the recombination returns the real function $T_X = T_1 \cdot T_R + T_2 \cdot (1 - T_R)$, where T_R is a random real function with codomain $[0, 1]$.

In simple terms, this geometric semantic crossover operator creates a program, T_3 , that returns as output a weighted sum of the outputs of its parents T_1, T_2 . This ensures that the operator is semantically driven and also that it is a geometric crossover on the semantic space, in the sense that when the parents and descendant are mapped onto the semantic space the descendant is bound to be between the parents and. Therefore, assuming that a target point in the semantic space exists the Euclidean distance between the descendant and that point is never larger than the distance of the farthest parent. Figure 1 illustrates this property. A formal demonstration and additional semantic geometric operators can be found in Moraglio et al. [14]

Geometric Semantic Mutation: Given a parent function $T : \mathbb{R}^n \rightarrow \mathbb{R}$, the mutation returns the real function $T_M = T + ms \cdot (T_{R1} - T_{R2})$, where T_{R1} and T_{R2} are random real functions and ms is the mutation step.

The use of two random real functions, T_{R1}, T_{R2} ensures that the semantic perturbation is centered around zero while the mutation step, ms , allows one to control the degree of the perturbations. Moraglio et al. [14] formally prove that this operator induces a unimodal fitness landscape, corresponding to a box mutation on the semantic space.

3. EVOLUTIONARY ART

The seminal work of Sims [18] created what is now the most popular approach to evolutionary art, the *expression-based* approach. Each genotype is a tree that encodes a symbolic expression. The rendering of the expression results in a phenotype, i.e., an image. The genotype-phenotype mapping process varies, but, in general, the phenotype can be seen as a visualization of the output of the s -*expression* over a set of variable values. Typically, evolution is user-guided although noteworthy exceptions exist [10, 17]. Some

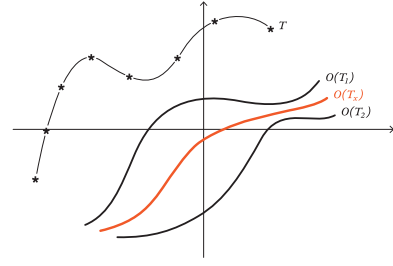


Figure 1: An illustration of geometric semantic crossover. The output of the descendant, $O(T_X)$, is bound to be between the outputs of the parents, $O(T_1), O(T_2)$, and, therefore, it is never farther from the target, T , than the worst parent, T_2 in this case. Adapted from Vanneschi et al. [21].

examples of this *expression-based* approach can be found in [18, 19, 22, 12, 7]. A thorough survey of EA systems beyond the scope of this paper (see, e.g., [10] for a survey).

As Machado and Cardoso [12] point out, most *expression-based* EA systems are theoretically able to create any image. Nevertheless, in practice, the image space that is actually explored depends heavily on the particularities of the system (primitives, genetic operators, genotype-phenotype mapping, etc.). McCormack [13] identified the problem of finding a symbolic-expression that corresponds to a known “target” image as one of the open problems of evolutionary art. More exactly, the issue is not finding a symbolic-expression, since this can be done trivially [12], the issue is finding a compact expression that provides a good approximation of the “target” image taking advantage of its structure.

In spite of the importance of this specific open problem, it is important to mention that in most evolutionary art applications there is no “target” image to obtain. The goals can be as varied as satisfying the user, as is the case of interactive evolutionary art, satisfy a hard-wired or learned aesthetic criteria [1, 5, 12, 4, 11], create novel images [9, 3], respond to the “infections” of parasites in a co-evolutionary scheme [6], etc.

Nevertheless, the use of semantic operators, may result in increased evolvability, thus promoting the fruitful exploration of the search space and potentially lead to better results, independently of the criteria used to judge the quality of those results.

Those familiar with Sims [18] influential work may recall that in addition to the evolution of static images Sims also evolved short animations. These animations were produced by a process called *genetic cross dissolve*, given two symbolic expressions, corresponding to two images, this process creates an animation by generating intermediate frames through the “interpolation” between the symbolic expressions over time. As we will see in Section 4, in its simplest form, this process is analogous to the geometric semantic operator presented in the previous Section. Additionally, Sims also mentions using this approach to mate individuals. Thus, evolutionary art adopted semantic methods since its early beginning, in fact, as far as we know, Sims [18] was the first to use semantic methods in GP. However, due to the limitations of the method, this contribution is often overlooked.

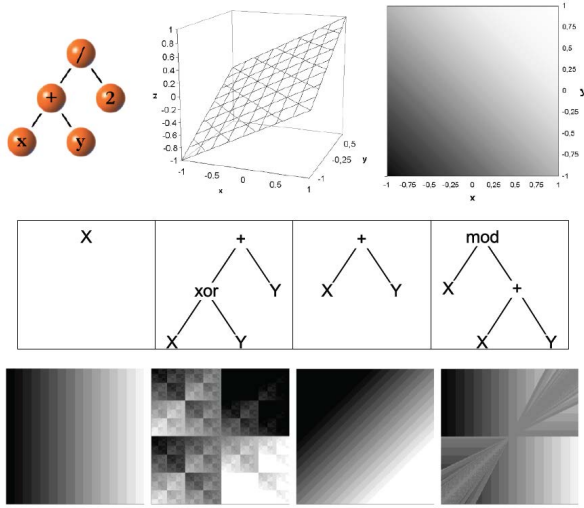


Figure 2: Representation scheme with examples of functions and the corresponding images.

3.1 Evolutionary Art Engine

The engine used in these experiments is inspired by the works of Sims [18]. It is a general purpose, expression-based, GP image generation engine that allows the evolution of populations of images. The genotypes are trees composed of 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 randomly initialized constants, these can be scalars or 3d vectors. The 3d vectors introduce color by specifying different values for the RGB color channels, e.g. $\#(0.1, 0.2, 0.1)$. Although a single tree is used for the 3 color channels, this tree can be decomposed into three different trees, one for each color channel, operating and returning real numbers. 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 (see figure 2). A thorough description of the GP engine can be found in [12]. Figure 3 displays typical imagery produced via interactive evolution using this system.

4. SEMANTIC METHODS FOR EVOLUTIONARY ART

In Section 2 we described geometric semantic crossover and mutation operators. Here, we describe our implementation of such operators and introduce novel semantic-inspired operators. These are based on the early work of Sims [18] and Hart [7] on the creation of short animations depicting smooth transitions between two evolved images.

The production of such animations relies on the use of the *lerp* function, which can be defined as follows:

$$\text{lerp}(A, B) = A \cdot t + B \cdot (t - 1),$$

with $A, B : \mathbb{R}^n \rightarrow \mathbb{R}$ and $t \in [0, 1]$.

To generate a video one creates a new individual $\text{lerp}(A, B)$ and calculates its output for different values of

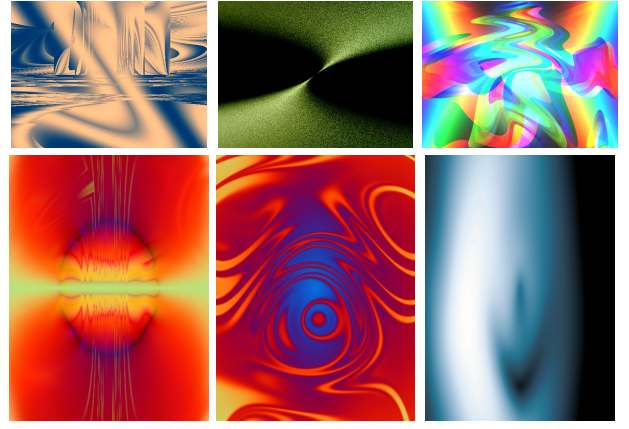


Figure 3: Examples of images generated by the evolutionary engine using interactive evolution.

t , starting with $t = 0$ and increasing t by a fixed step until $t = 1$, which produces a “fade” between A and B . In most cases, this results in rather uninteresting videos since it is simply a static morph between two images. Sims [18] proposes an approach based on *first-differences* alignment which allows him to obtain swift transitions between pairs of images while introducing the illusion of movement and change over time. Later Hart [7] refined this early approach introducing novel alignment schemes which extend the range of results producible through genetic cross dissolve.

The similarity between the *lerp* function and the geometric semantic crossover defined in Section 2 are striking. In fact, the only difference is the following: *lerp* uses the variable t to control the influence A and B on the outcome while the geometric semantic crossover uses a random function, T_R , with codomain $[0, 1]$ for the same effect.

To make them equivalent we redefine *lerp* as follows:

$$\text{lerp}(T_1, T_2, T_R) = T_1 \cdot \text{sig}(T_R) + T_2 \cdot (1 - \text{sig}(T_R)),$$

with $T_1, T_2, T_R : \mathbb{R}^n \rightarrow \mathbb{R}$, and $\text{sig}(x) = \frac{1}{1+e^{-x}}$.

Following the work of Vanneschi et al. [21], we use *sig* to map \mathbb{R} into $[0, 1]$, thus removing the constraint of using as third argument a function with codomain $[0, 1]$.

In the next sections we describe several genotype alignment techniques and the corresponding genetic operators.

4.1 Crossover and Tree-Alignment

The first crossover operator described herein is similar to the semantic crossover operator presented in Section 2. The second results, directly, from the work of Sims [18], who used a similar technique for creating animations and performing crossover between parents with similar structures. The last two crossover operators are inspired on the work of Hart [7]. The difference between the four operators lies on the algorithm for aligning the genotypes, i.e. finding matches between pairs of genotypes [15, 16].

4.1.1 Root Alignment

This is the simplest alignment technique, the roots of both parents, A and B are aligned and no further alignments are made. Considering this technique the result of the crossover operator between any pair of individuals A and B is $\text{lerp}(A, B, T_R)$, where T_R is a randomly generated

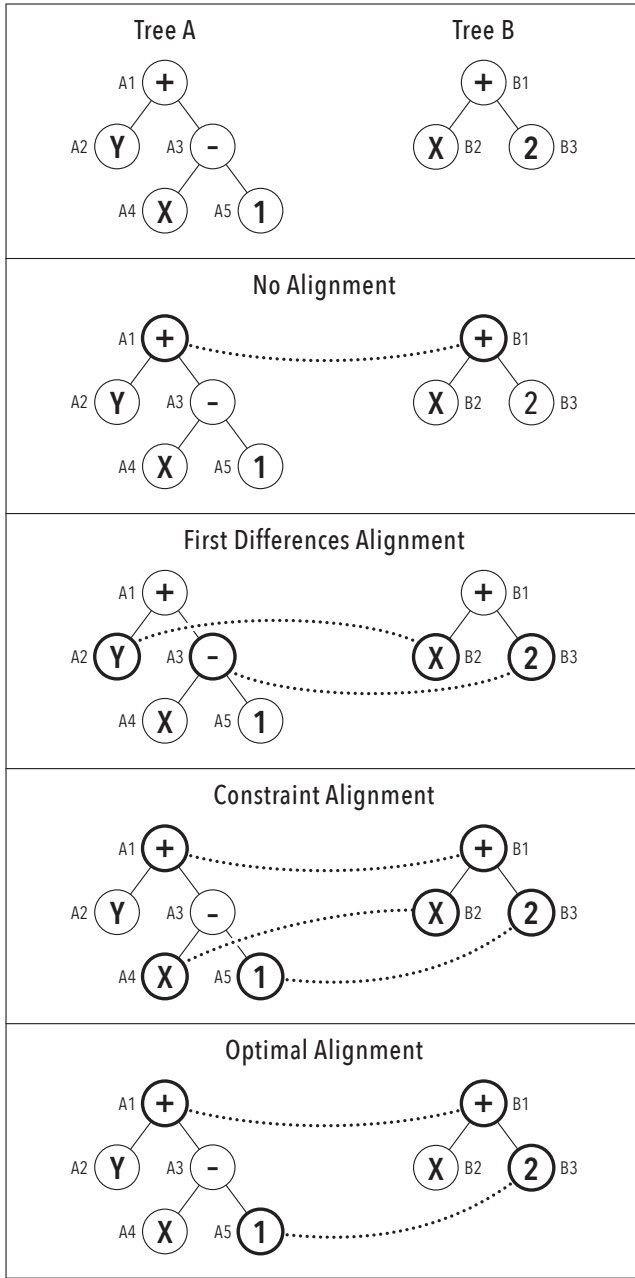


Figure 4: Alignments of A and B according to different algorithms.

tree. Thus, the crossover between the individuals depicted in figure 4 would result in: $lerp(+ (y, - (x, 1)), + (x, 2), T_R)$. As previously mentioned, this operator is equivalent to the one presented in Section 2, sharing the exact same properties, and can, therefore, be classified as a geometric semantic operator.

4.1.2 First Differences Alignment

The first difference alignment algorithm can be described as follows: If the roots of the trees are equal call the alignment algorithm for each of their arguments (unless they are leaves); if the roots are different align both roots.

Running this algorithm for the example presented in figure 4 produces the following alignments: $(A2, B2)$ and $(A3, B3)$. Like previously, to produce the descendant one *lerp* node is introduced for each alignment, resulting in: $+ (lerp(x, y, T_{R1}), lerp(-(x, 1), 2, T_{R2}))$.

Although the operator is semantic-inspired, it can no longer be classified as a geometric semantic operator since the descendant is no longer necessarily “between” the two parents. In other words, considering a target image T and Euclidean distance, while *root alignment* crossover returns a descendant that is at least as close to T as the farthest of its parents, *first difference alignment* crossover may return a descendant that is farther from T than the farthest parent.

This approach was used by Sims [18] for creating his famous *genetic cross dissolves* between images. If the two expressions have different root nodes *first differences* and *root alignment* produce the same outcome. In Sims work, the resulting video would be a “traditional”, and rather uninteresting, fade between the images. However, when the trees share a common structure (i.e., when they have at least the root node in common) interesting movements and unexpected, yet smooth, transitions tend to occur.

Since Sims [18] and Hart [7] were primarily focused on the production of videos, they control the influence of each parent on the outcome through the variable t . We use a random function for the same purpose, which introduces a subtle, but important, difference between the approaches. In the original version the relative influence of the parents is homogeneous across the image since it only depends on t , however, when a random function – which can use variables x and y , and as such return different outputs for different x, y coordinates – is used this is no longer true, e.g. the upper part of the image may be strongly influenced by the first parent while the bottom part is mainly influenced by the second.

4.1.3 Constraint Alignment

As noted by Hart [7], two genotypes are likely to have different root nodes, and the result of the previously described operators is likely to be a rather uninteresting blend of the two images. Thus, although *root alignment* may be a valuable operator for “conventional” GP applications such as symbolic regression, its interest in the context of evolutionary art appears rather limited. To some extent, the same applies to *first differences* alignment since, in most cases, it will give the same output as *root alignment*. Considering that it may be desirable to “continue matching nodes after encountering a difference between the trees, since the result of the blend will often move rather than fade” [7], Hart proposes two additional alignment techniques. Like *first differences* alignment, the resulting crossover operators are semantic-inspired, but cannot be classified as geometric semantic operators under Euclidean distance.

The first technique is *constraint alignment* which can be informally described as follows: Considering two trees, A and B , we begin by establishing a correspondence between the levels of the trees; since the trees may have different heights, we will randomly discard levels of the larger tree, so that the number of levels to be considered is the same (the root level is never discarded); the remaining levels are then matched in top to bottom fashion (and hence there is always a match between the two root levels); we then proceed in level by level fashion, randomly discarding nodes

so that the number of nodes at each level coincides; the remaining nodes are aligned in left to right order.

In figure 4 we present a possible outcome of the *constraint alignment* crossover between two individuals. Parent A has a height of 3, B has a height of 2. Therefore, for alignment purposes, it is necessary to randomly discard a level of parent A (either level 2 or 3, since the root is never discarded). In this case level 2 was discarded, thus the first level A is matched with the first of B and the third level of A is matched with the second of B . Since the number of nodes per considered level is the same for both parents, it is unnecessary to discard further nodes. Finally the nodes are aligned per level from left to right resulting in the following alignments: $(A1, B1)$, $(A4, B2)$, $(A5, B3)$.

As previously, each alignment results in the introduction of *lerp* operators. Therefore the crossover of A and B results in: $lerp(+ (y, -(lerp(x, x, T_{R1}), lerp(1, 2, T_{R2}))), +(lerp(x, x, T_{R3}), lerp(1, 2, T_{R4})), T_{R5})$.

4.1.4 Optimal Alignment

Based on the work of Jiang et al. [8] on the alignment of ordered phylogenetic trees, Hart [7] proposes an optimal alignment scheme. This algorithm calculates all possible alignments between pairs of nodes and forests of children, that is every node of the first parent is compared against every node of the second parent and every forest of children of a node of the first parent is compared against every forest of children of every node of the second. As is demonstrated by Jiang et al. [8], the algorithm they propose has a quadratic temporal complexity. The quality of each alignment between a pair of nodes, n_1, n_2 is determined by a function, μ , which penalizes alignments between nodes of different types. Hart [7] adapted μ to a GP context, proposing the following function: $\mu(a, b) = 0$ when $a = b$; $\mu(a, b) = 1$ when $a \neq b$ but a and b are of the same type (i.e., both are variables, constants, or functions); $\mu(a, \lambda) = mu(\lambda, a) = 1$ that is not aligning a node gives a penalty of 1; $\mu(a, b) = 2$ when $a \neq b$ and a and b are of different types. The quality of an alignment between two trees is given by the sum of the μ values of all alignments between nodes. The algorithm efficiently calculates an alignment that minimizes μ . There is always an alignment between the root nodes of the parents.

A thorough description of the optimal alignment algorithm is beyond the scope of this paper, so we redirect the reader to the original works of Jiang et al. [8] and Hart [7].

For the example presented in figure 4 the optimal alignment between parent A and B is $(A1, B1)$ and $(A5, B3)$. As such the result of the optimal alignment crossover operation is: $lerp(+ (y, -(x, lerp(1, 2, T_{R1}))), +(x, (1, 2, T_{R2})), T_{R3})$.

4.2 Mutation

Our mutation operators take full advantage of the tree alignment methods that were previously described. Therefore we have *root*, *first differences*, *constraint* and *optimal* alignment mutations. All of these operators can be described as follows: $mutation(A) = crossover(A, B.ms)$, with $A, B : \mathbb{R}^n \rightarrow \mathbb{R}$, where B is a random function and ms is the mutation step. In other words to mutate an individual we perform a crossover between this individual and a random tree, using a specific alignment scheme.

For *root* alignment, this mutation operator is similar to the geometric mutation operator described in Section 2, since the use of the *sig* function ensures a codomain of

$[0, 1]$ and since the outputs of our random trees are naturally centered around zero. Therefore, *root* alignment mutation is also a geometric semantic operator under Euclidean distance. However the same cannot be stated for the remaining mutation operators.

5. EXPERIMENTATION

The use of semantic genetic operators raises a practical problem, the growth of program size is so accentuated that, after a few generations, the individuals become so large that it becomes impossible to calculate their output by transversing the program tree. In some scenarios, it is possible to calculate the output of a program from the outputs of its parents [21], which circumvents this specific problem. However, this is only possible for the *root* alignment operators. When other types of alignment are used it would be necessary to use dynamic programming techniques to calculate and store intermediate results. Although these techniques postpone the problem for a while, they do not solve it. The same applies to the simplification of the symbolic expressions proposed by Moraglio et al. [14].

For these reasons performing long evolutionary runs is not feasible. As such we consider that, at least in the context of evolutionary art, the most promising applications of these semantic-inspired operators are the following:

1. Short evolutionary runs with large populations;
2. Long evolutionary runs using conventional operators by default and semantic operators on rare occasions;
3. Semantic recombination of highly fit individuals evolved using conventional operators;
4. Semantic mutation of highly fit individuals evolved using conventional operators;

5.1 Runs With Semantic-Inspired Operators

We conducted a series of interactive evolutionary runs using the proposed semantic-inspired mutation and crossover operators. In each run we selected one alignment scheme – *root*, *first differences*, *constraint*, *optimal* – and used the associated crossover and mutation operators. The experimental settings were the following: population size = 100; number of generations 10; crossover probability = 0.8; mutation probability = 0.2; tournament selection, with tournament size = 5; elitist strategy, with elite size = 1.

When we compare the results obtained using semantic-inspired operators with those obtained using conventional operators one can observe a striking difference: the use of semantic operators results in a dramatic reduction of population diversity. This is somewhat expected since geometric semantic crossover ensures that the descendants are “between” both parents (see Section 2). Therefore, in the absence of mutation, search is constrained to the regions between parents. After a few populations selection pressure guides evolution towards an even smaller sub-region of the space. Geometric semantic mutation may introduce novelty in the population. However, in the experimental conditions, the mutations are unable to introduce enough novelty to keep the runs sufficiently interesting, even when mutation steps, ms , as high as 1 are considered. The loss of diversity is particularly visible when using *root*, which is a geometric semantic operator, and *first differences* alignment,

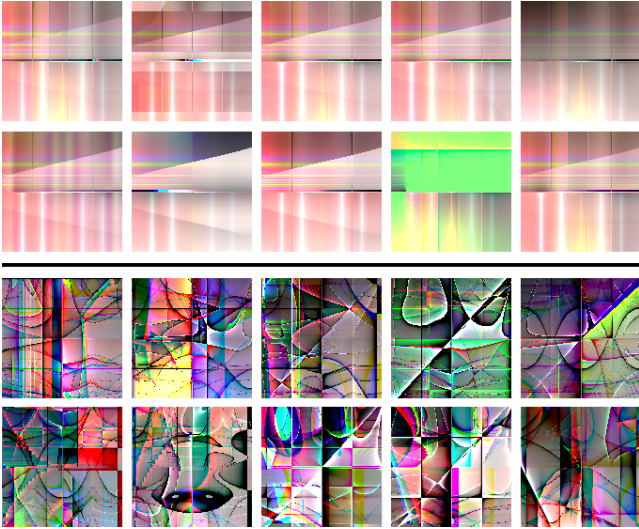


Figure 5: Partial screenshots of populations from runs using root (top two rows) and constraint (bottom two rows) alignment.

which behaves similarly to *root* alignment when the roots of the parents are different. Conversely, *constraint* and *optimal* alignment, which although semantic-inspired are not geometric operators, are able to sustain diversity for longer periods of time. In terms of user experience the runs using semantic-inspired operators can be considered largely disappointing, due to the lack of diversity, and also because the users feel, and to some extent are, committed to the choices made in the first few generations and unable to steer evolution in a different direction. In figure 5 we present partial snapshots of the 15th population of two runs using *root* and *constraint* alignment, highlighting the differences in diversity between the results.

We also conducted several runs using a hardwired fitness function similar to the one presented by Machado and Cardoso [12]. In terms of population diversity the results are comparable to those reported for interactive evolution runs. However, looking at these runs exclusively from an optimization point of view the differences between standard and semantic-inspired operators become less visible. That is, fitness increases along the evolutionary process and the differences of performance among operators are not statistically significant. Our explanation for this result is the following: although the theoretical properties of geometric semantic operators should give them an edge in optimization tasks, the runs are short and so these advantages are not visible.

5.2 Semantic Manipulation of Fit Individuals

The nature of semantic-inspired genetic operators and the quality of the *genetic cross dissolves* presented by Sims [18] and Hart [7] indicate that they can be particularly well-suited for recombining and mutating fit individuals. We conducted a series of tests using fit individuals evolved previously through interactive evolution using conventional genetic operators. These individuals were then recombined and mutated using semantic-inspired operators.

Figure 6 depicts typical examples of the recombination of individuals with similar structures, in particular with the same root node. As it can be observed, conventional

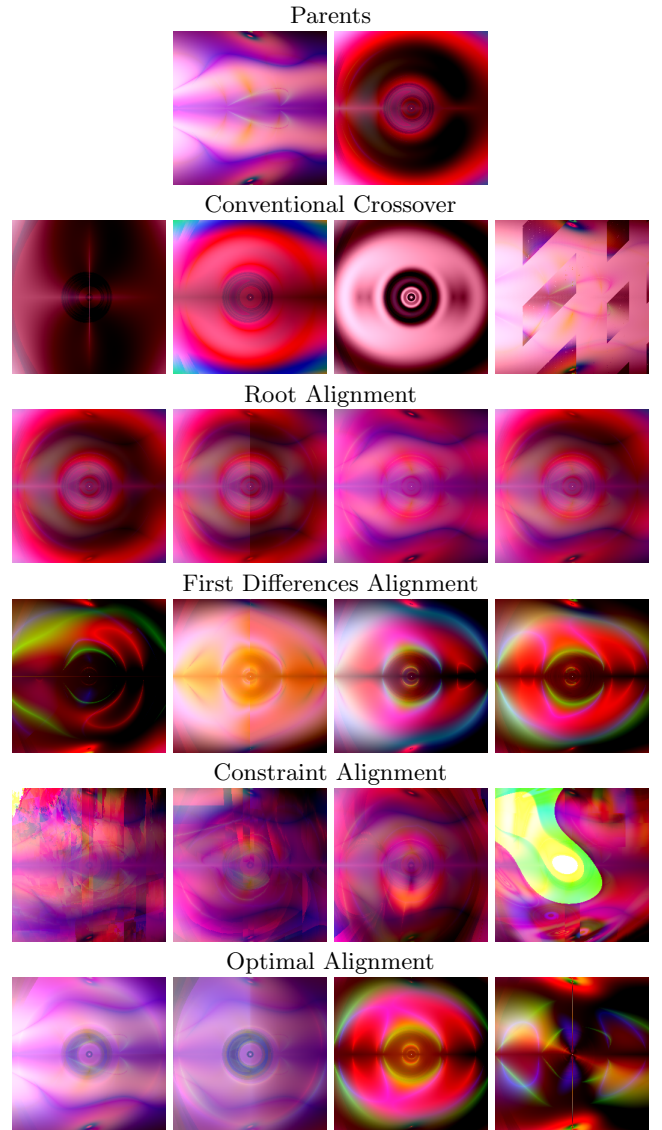


Figure 6: Examples of the recombination of individuals with similar structures.

crossover produces a wide variety of outputs that may show strong deviations from the parents, while *root* alignment produces fades between the two parents and, therefore, predictable results. *First differences* alignment produced a wide range of interesting results that, in our subjective opinion, appear to be a good compromise between diversity and inheritance of visual characteristics from the parents. The same can be stated regarding *optimal* alignment, although the results tend to be less varied. *Constraint* alignment produced unexpected results, it often produces descendants that close matches to one of the parents, fades between parents and totally unexpected images that bare little resemblance to any of the parents. Our explanation for this behaviour is the following: the alignments made using the *constraint* technique are largely arbitrary and so are the results. Considering the overall results, we find that semantic-inspired operators are valuable for the recombination of structurally similar individuals, and that *first differences* and *optimal* alignment tend to produce the best results.

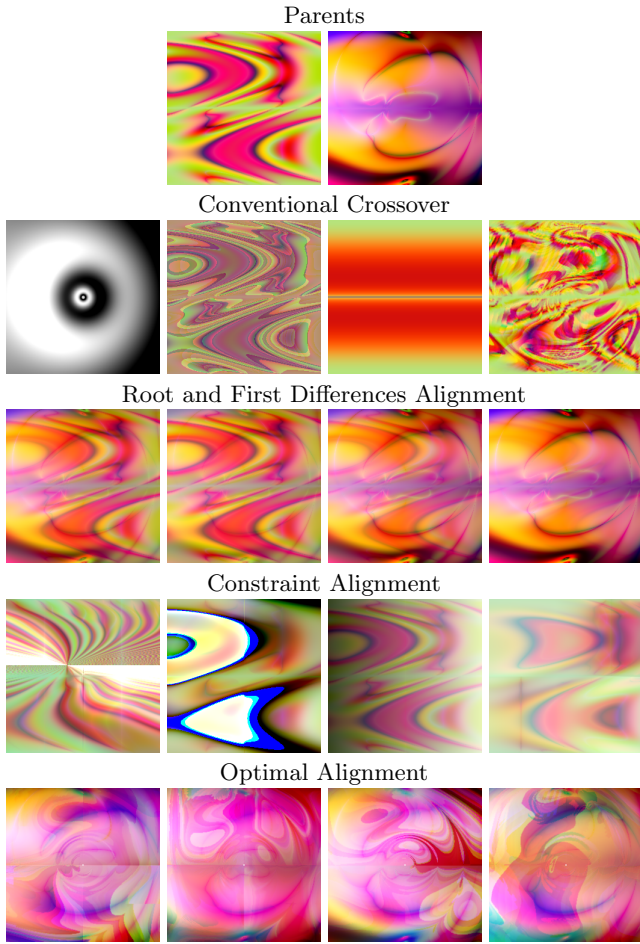


Figure 7: Examples of the recombination of individuals with dissimilar structures.

Figure 7 presents varied examples of the recombination of individuals with dissimilar structures. Since the root nodes are different *root* and *first differences* alignment are equivalent. As previously, conventional crossover produces a wide range of outputs and strong deviations from the parents are, apparently, more common than when the individuals have similar structures. *Root* alignment disregards the structure of the individuals and, as such, we obtain the same type of fades between parents that can be observed when recombining structurally similar individuals. The erratic behaviour of *constraint* alignment is accentuated and the individuals tend to fall on one of two extremes: a fade between the parents, an image that is apparently unrelated to any of them. *Optimal* alignment shows, by far, the highest percentage of recombinations that simultaneously inherit visual characteristics from both parents and depict some novelty. For this reason we considered it the most useful operator for the recombination of structurally dissimilar parents.

Figure 8 shows typical examples of the mutation operator. In general, the statements made regarding the crossover of individuals with different structures also apply to mutation. This is an expected result since, in essence, the semantic-inspired mutation operators perform the crossover of an individual with a randomly generated tree. The differences among semantic-inspired operators tend to be less visible.

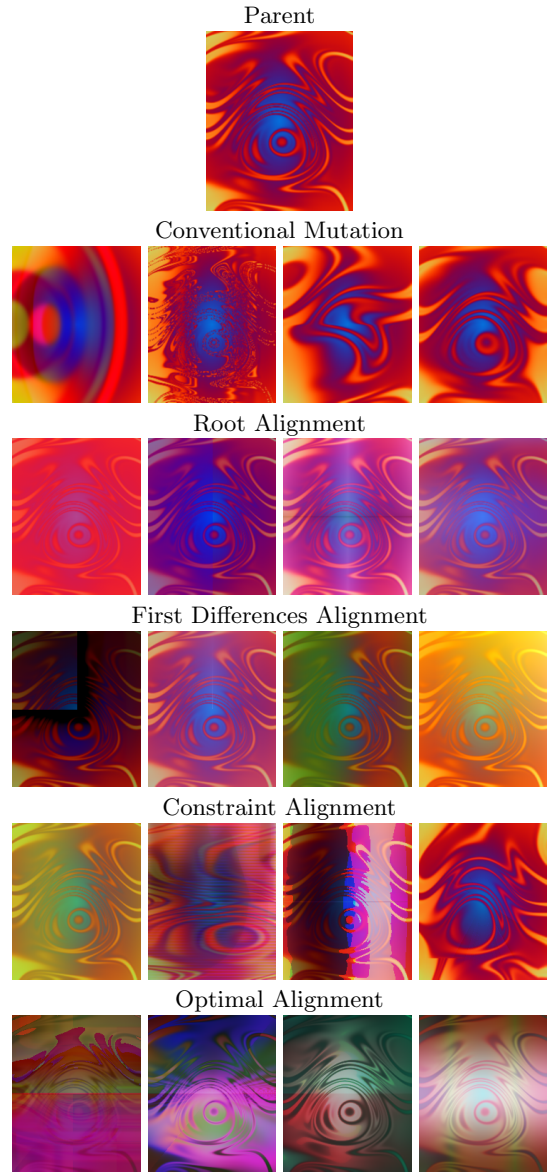


Figure 8: Examples of mutation operations.

Since the randomly generated trees are significantly smaller than the individuals used for crossover, the opportunities for alignment are scarcer, which dilutes the differences. Although semantic-inspired mutations may be useful for introducing subtle changes, in general, and in our subjective opinion, they offered little or no advantage over conventional mutation. In particular, they all tend to fail to systematically introduce novelty into the population and thus to keep the users interested in the evolutionary process.

6. CONCLUSIONS

We presented several semantic-inspired genetic operators using different genotype alignment techniques and applied them in the context of evolutionary art. The analysis of the experimental results indicates that, while these operators have practical limitations due to the exponential increase of program size, they are useful for specific tasks. In partic-

ular these semantic method are promising for the recombination of particularly fit individuals. On the other hand, performing long evolutionary runs, particularly interactive ones, appears to be unfeasible and tedious.

In the future we intend to conduct several interactive and automated runs using a combination of conventional and semantic operators. This combination may be useful for avoiding the lack of population diversity and the exponential growth of program size induced by the use of semantic operators, but still preserve their advantages. To achieve this, the semantic operators should be used seldom, which implies that identifying the occasions where their use is advantageous becomes an important research question.

7. ACKNOWLEDGMENTS

This research is partially funded by: iCIS project (CENTRO-07-ST24-FEDER-002003), which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER; Fundação para a Ciência e Tecnologia (FCT), Portugal, under the grant SFRH/BD/90968/2012; project ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

8. REFERENCES

- [1] S. Baluja, D. Pomerlau, and J. Todd. Towards automated artificial evolution for computer-generated images. *Connection Science*, 6(2):325–354, 1994.
- [2] L. Beadle and C. G. Johnson. Semantically driven mutation in genetic programming. In *IEEE Congress on Evolutionary Computation*, pages 1336–1342. IEEE, 2009.
- [3] J. Correia, P. Machado, J. Romero, and A. Carballal. Feature selection and novelty in computational aesthetics. In P. Machado, J. McDermott, and A. Carballal, editors, *EvoMUSART*, volume 7834 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 2013.
- [4] A. Ekárt, A. Joó, D. Sharma, and S. Chalakov. Modelling the underlying principles of human aesthetic preference in evolutionary art. *Journal of Mathematics and the Arts*, 6(2-3):107–124, 2012.
- [5] G. Greenfield. Evolving aesthetic images using multiobjective optimization. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 3, pages 1903–1909 Vol.3, 2003.
- [6] G. R. Greenfield. On the co-evolution of evolving expressions. *International Journal of Computational Intelligence and Applications*, 2(1):17–31, 2002.
- [7] D. A. Hart. Toward greater artistic control for interactive evolution of images and animation. In *Proceedings of the 2007 EvoWorkshops 2007 on EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog*, pages 527–536, Berlin, Heidelberg, 2007. Springer-Verlag.
- [8] T. Jiang, L. Wang, and K. Zhang. Alignment of trees – an alternative to tree edit. *Theoretical Computer Science*, 143(1):137–148, 1995.
- [9] T. Kowaliw, A. Dorin, and J. McCormack. An Empirical Exploration of a Definition of Creative Novelty for Generative Art. In K. B. Korb, M. Randall, and T. Hendtlass, editors, *ACAL*, volume 5865 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2009.
- [10] 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.
- [11] Y. Li, C. Hu, L. L. Minku, and H. Zuo. Learning aesthetic judgements in evolutionary art systems. *Genetic Programming and Evolvable Machines*, 14(3):315–337, 2013.
- [12] P. Machado and A. Cardoso. All the truth about NEvAr. *Applied Intelligence, Special Issue on Creative Systems*, 16(2):101–119, 2002.
- [13] 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.
- [14] A. Moraglio, K. Krawiec, and C. G. Johnson. Geometric semantic genetic programming. In C. A. C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *PPSN (1)*, volume 7491 of *Lecture Notes in Computer Science*, pages 21–31. Springer, 2012.
- [15] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.
- [16] C. Notredame. Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1):131–144, 2002.
- [17] J. Romero and P. Machado, editors. *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Natural Computing Series. Springer Berlin Heidelberg, November 2007.
- [18] K. Sims. Artificial evolution for computer graphics. *ACM Computer Graphics*, 25:319–328, 1991.
- [19] T. Unemi. Sbart2.4: Breeding 2d cg images and movies, and creating a type of collage. In *The Third International Conference on Knowledge-based Intelligent Information Engineering Systems*, pages 288–291, Adelaide, Australia, 1999.
- [20] N. Q. Uy, N. X. Hoai, M. O'Neill, R. I. McKay, and E. G. López. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, 2011.
- [21] L. Vanneschi, M. Castelli, L. Manzoni, and S. Silva. A new implementation of geometric semantic gp and its application to problems in pharmacokinetics. In K. Krawiec, A. Moraglio, T. Hu, A. S. Etaner-Uyar, and B. Hu, editors, *EuroGP*, volume 7831 of *Lecture Notes in Computer Science*, pages 205–216. Springer, 2013.
- [22] L. World. Aesthetic selection: The evolutionary art of steven rooke. *IEEE Computer Graphics and Applications*, 16(1), 1996.