Producing Prouns: Building a genetic algorithm while respecting strict aesthetic guidelines

Robert Gerdisch

Abstract
This paper investigates the challenges associated with building an algorithm for procedurally generating artificial creatures that fulfill a functional goal while also satisfying artistic design constraints. The algorithm achieves this through a novel combination of previously disparate strategies used in other genetic algorithm implementations, effectively modernizing early foundational work. One strategy in particular, active abnormality detection, will prove particularly useful in balancing form exploration with functional stability, and provide the algorithm with its novelty. The end result is an algorithm capable of producing creatures demonstrating pseudo-ecological aesthetic diversity, while stably satisfying guidelines set out by the original designer of the objective function and the physical design of the creatures. Here we will explore a case study where this was used directly to produce satisfactory results, and discuss possible improvements to the proposed algorithm. It will also be shown that a genetic algorithm can be built in order to leverage its ability to produce controlled aesthetic diversity instead of for optimizing a carefully-constructed goal function. The genetic algorithm described by this paper focuses less on traditional fitness performance and more on producing an output that is visually viable to the end-user. This allows the majority of the implementation work to be put towards fine-tuning the interaction between the genome specification and the stylistic preferences of the designer, and finding an algorithmically representable way to identify and remove generated organisms that violate either the aesthetic guidelines or the constraints imposed by the physics simulation governing the organisms' movements in a virtual environment.
1 Introduction

Genetic algorithms are an excellent method for traversing a continuous or vaguely-defined search space, looking to build a solution that maximizes an objective of ranging complexity. Most often, genetic algorithms are simply provided a genome, a method for selection based on fitness, a crossover function, and sometimes probabilities for mutation of certain genes. Assuming the objective function and hyperparameters are tuned well, this can yield good results in a variety of domains.\footnote{Optimisation of Energy and Exergy of Turbofan Engines Using Genetic Algorithms} However, by the nature of their freedom, most genetic algorithm implementations can find solutions that maximize the objective function but hold little to no regard for subtle aesthetic concerns that humans may crave over perfect performance in the objective.

Such is the case for art. An artist might be inclined to use a genetic algorithm to generate many distinct versions of a general idea, and design the objective function to some approximation of what they would like to see output. Yet while most of the functional qualities of gene expressions in the artificial evolution algorithm are trivially mathematically expressible for the purpose of comparison between different genomes, there are many qualities that are challenging to express reliably in a mathematical language. One such quality that is particularly challenging is measuring the stability of an artificial organism’s physical configuration within the rules of the physics simulation resolving the kinematics if the organism’s movement. “Exploding Prouns,” as they will come to be known, were a central issue in the case study described in section 3. Therefore, this paper will be have three main focuses: (1) building a modern genetic algorithm that leverages best practices both new and old, (2) discussing an implementation for detection of an aesthetic guideline violation for which a mathematical description is not readily apparent, and (3) proposing solutions to the nontrivial challenges that arose when building the algorithm and that remain unsolved.
2 Related Work

Karl Sims’ work is well-known and extremely popular among those who have any exposure to genetic algorithms. He laid the foundation (both technological and philosophical) for how they should work, and how they should be applied to generating virtual creatures.² We build upon his ideas here to generate virtual creatures, which provides a starting point for automating the exploration of complex design spaces with abstract (albeit efficient) controls for the designer. However, Sims’ approach takes care to maintain a certain degree of biological faithfulness that would provide unnecessary complexity if directly applied to this project. Our algorithm borrows a few of the more salient aspects of his work: using graphs to represent creature morphology, mating those graphs, measuring fitness with a physical simulation, and conceptually separating creature morphology from control while working to improve both of them simultaneously.

We will iterate on Sims’ graph genome by borrowing ideas from Evolving Neural Networks through Augmenting Topologies (NEAT).³ This implementation works similarly to Sims’ evolution—it aims to improve graph topologies (neural networks) while simultaneously improving the control structure (network weights). However, the algorithm adds two important concepts independent of the output domain: dormant genes and generational history. As evolution occurs over generations, new connections are added to the networks in the gene pool. As new sub-topologies develop, the connections being replaced are made dormant and the new connections are assigned “global innovation numbers”, which can be used to track any organism’s genetic history across generations. When mating two organisms at the end of a given generation, it is then possible to measure their relatedness by counting their matching genes. Furthermore, species can be identified (and protected) by grouping organisms that

share a certain number of genes. Speciation can be particularly useful for balancing diversity with evolution on a certain structure that is evolutionarily viable.

Finally, videos of DeepMind’s evolved locomotion behaviors are quickly becoming as popular as Karl Sims’ original videos. This technique involves a completely different approach to artificial evolution, focusing primarily on large scale reinforcement learning as an alternative to a traditional genetic algorithm implementation. While this method seems to work well for the problem DeepMind is looking to solve, it works solely to optimize the control structure of an explicitly designed morphology (i.e. a bipedal or quadrupedal organism, in one case a human and another a simplified spider model). This paper, like Sims’ work and NEAT, will also need to evolve on morphologies. However, the results from DeepMind were useful in providing inspirational examples demonstrating the potential power of evolutionary algorithms for generating physically plausible animations for models in 3D.

---

3 Prouns


Originally imagined by El Lissitzky in the early 20th century, Prouns are abstract representations of pre-war German architecture. Their “bodies” are composed of semi-orthogonally joined geometric primitives. Hito Steyerl, an artist of the 21st century, was inspired by both his art and the findings at DeepMind to attempt animation of these forms in 3D. She aimed to build a zoo full of them, and wanted the animations to be as realistic to their potential movements as possible. It would be a daunting task to attempt satisfactory animations for all of them, and so she wished to procedurally generate both animations and novel physical designs for each of them, in order to create an artificial ecosystem of Prouns. Completely random animations of joints might work, except it would likely not produce animations resembling creatures that crawl around a scene. Small random variations in an explicitly-designed animation might also work, but the variation in generation would be constricted. Therefore, a genetic algorithm seemed to be the best method for producing these animations.

---

4 Algorithm

Figure 2: A few Prouns generated by the genetic algorithm. See supplementary materials for the source video.

We construct the virtual Prouns as an undirected graph of nodes. These nodes are mostly boxes, like those that can be seen in Lissitzky’s work. Some other shapes like sticks and circles are also used to improve visual diversity within the creatures. Invisible joints and muscles connect the nodes, and they either fix the nodes to each other or contract periodically along their axis to provide a springing energy. All of the objects that make up the body and “mind” of the Proun are placed along orthogonal axes within the Proun’s local coordinate system. This gives a nice approximation of the orthogonality that dominates Lissitzky’s style, and the orthogonality is not strictly preserved in motion, so the Prouns still resemble Lissitzky’s paintings while also resembling biological creatures when they are crawling, bouncing, and rolling around the scene.
4.1 Genome Design

The genome is a serialization of the graph that defines the Prouns. As was hinted at above, the genome is separated into “mind” and “body.” The clear separation is important for our purposes, as it will make simultaneous form and function evolution more intuitive. With this method, we can more cleanly fix the control structure as the nodes mutate, and vice versa. As will be described in section 4.3, we also have different algorithms for mating minds and bodies.

The genes defining the nodes that make up the Proun’s body have parameters for mass, spatial dimensions, friction against the ground plane, color, and a type, which in the current implementation resolves to a box, a stick, or a circle. Aside from color, all of these parameters have direct implications on the physical method by which Prouns navigate the scene, similar to the differentiation seen in Sims’ work. Some Prouns are lighter with more active muscles and so they bounce around the scene like pogo sticks, other Prouns have especially round bodies and roll around the scene, and yet others are more top-heavy, so they crawl around the scene.
Figure 4: Two small Prouns next to each other. One is composed of sticks and boxes and the other of a stick and circles.

All of the genes that define the Proun’s mind have parameters defining the nodes they connect, the orthogonal axis along which they are aligned (the x, y, or z axis of the Proun’s local coordinate system), and a type (either fixed or moving). If the muscle is capable of movement, its gene has parameters for the period of the movement, the contraction distance, and the full-length extended distance. The muscles seem to have less of an impact on the movement of the Prouns than the nodes, but as we will see in section 4.4 they are the most fragile part of the system. Small variations can quickly lead to Prouns which are either too “weak” to move or their muscles are too enmeshed to work properly in concert.
4.2 Selection

The function that calculates fitness for a Proun. The unnamed numbers are constants that were tuned through experimental trial and error to produce informative fitness metrics.

The objective function that drives the Prouns’ evolution is composed of three parts: complexity, travel, and stability. The “complexity fitness” of a given organism is measured by multiplying the cube of the number of nodes in the Proun’s body with the Proun’s muscle density, which is calculated as \( \frac{2 \times \text{number of muscles}}{\text{number of nodes}} \). The total number of nodes and muscles in a Proun is globally limited by a parameter set by the designer, so the fitness function does not have to create its own artificial ceiling for the complexity of a Proun. Instead, more complicated Prouns that successfully traverse the scene (denoted by total displacement recorded at the time of “death”) are rewarded for exhibiting complexity while remaining stable. Finally, the stability metric measures and penalizes the number of times the Proun exhibited aberrant behavior. In this case, aberrant behavior means either moving too quickly or flying too high in the air. This is discussed more in-depth in section 4.4, however these two examples of behavior typically correspond to muscular configurations that do not generate aesthetically pleasing or even biologically plausible movement.
Figure 6: The two methods of Proun selection used together to choose the next generation. In ShotgunSelect, blastSize is set to 3. In SelectTopGenomes, numTop is set to any number between 2 and 5.

After calculating fitnesses, Prouns are selected by of shotgun tournament-style selection, where two “shots” of Prouns are randomly chosen at a time and the fittest from each small group are the ones to mate. A few of the top Prouns are also chosen as “elites” and asexually reproduce to the next generation in order to preserve high fitness functions across generations. There are many competing methods that have benefits and consequences for selecting specimens for reproduction into the next generation, but this method was chosen as it is being particularly useful for selecting a wide range of fitnesses that are also often better relative to the average of the previous generation. Elitism is a common practice among most artificial evolution algorithms and does a good job of preserving good examples that are found in intermediate generations that will likely be used by the designer in the final result.
4.3 Mating

4.3.1 Crossover

In order to build Prouns from the previous generation in order to fill the next generation, we use the crossover method, which aims to mix genomes in a way conceptually analogous to how parent genomes contribute to their offspring’s genome through biological sexual reproduction. There are two areas in which our method of genome crossover is unique. The first is in selecting the genes that will go into the child. The original implementation sliced the parents and combined random halves, which is fairly standard among genetic algorithm implementations, and is the most intuitive way to capitalize on sub-solutions (within the parents) to build a more complex and likely fitter solution (the child). However, this method of crossover did not promote the degree of diversity we sought from the final output. Instead, we do a series of weighted coin flips for every gene index shared by the child’s parents. For example, if parent A is composed of 8 nodes and the parent B has 12 nodes, then the child will start by containing 8 nodes, via weighted randomness on both parents’ nodes at indices 0-7. The randomness is weighted by the proportional difference in fitness between the two parents. For example, if parent B is twice as fit as parent A, then the child is twice as likely to inherit genes from parent B. Finally, if the fitter parent is also larger, then the child inherits all of the fitter parent’s extra nodes. In our example the child would receive parent B’s nodes at indices 8-11. If the fitter parent is smaller, the child receives no extra nodes.

The second area is how the control structure is passed to the child. The muscles from the less fit parent are first transferred onto the topology of the child, but these genes are immediately set to dormant, therefore they will have no perceivable effect on the child and can only be reactivated through mutation in a future generation. The fitter parent then overwrites the control structure passed to the child in the first step with its own muscles, but they are left in their original state, either active or dormant from a
previous generation. This allows us to preserve diversity while also actively working towards a fitter organism.

4.3.2 Mutation

Beyond affecting small, random nudges within the genetic parameters of the Proun, the mutation step also has a chance of activating dormant genes or deactivating active genes. We also added a third state for muscular activity, limp, which causes the muscle to exert a weak binding force on the nodes it joins, but otherwise does not actively contribute to the movement of the Proun besides offering extra cohesion to the Proun's body. See figure 7 for details on how the mutation probabilities in muscles are controlled. Since nodes are less likely to contribute to Proun destabilization, their mutation parameters are given less attention, and can be tuned liberally according to general design preferences.

---

**Figure 7:** The control panel for mutation probabilities and maximum magnitudes in Proun muscles. First we have an overall probability for mutation, which is typically kept just over 50%. Each mutation on individual aspects of the child's muscles have unique probabilities of occurring, assuming the muscle has passed the overall probability of mutation. These values have been tuned to balance rate of aesthetically significant evolution with overall stability. The probability for gene activation and deactivation, a mutation labeled “enable mode,” is kept low because this mutation, while powerful for evolution, can easily cause destabilization, as it can introduce unintentionally dense muscular topologies.
In the first implementation, the generated Prouns simply did not work. In very simple or straightforward cases, such as three or four nodes connected by two or three muscles, the Prouns behaved well. However, any additional complexity would cause extremely erratic behavior—diverging vibration, rapid spinning in place—most often ending in explosion-like behavior (see above). This is not at all the type of aesthetic result we were hoping to achieve. One or two aberrant Prouns appearing in the gene pool threatened to spread their corrupt genomes, permanently derailing the path of evolution towards total disarray.

This discovery posed the most significant challenge when building the algorithm. Due to the aesthetic constraint of “complex creatures whose internal parts move quasi-orthogonally with respect to each other,” the genetic algorithm was being encouraged to generate organisms with physical configurations that conflicted with
expectations imposed by the equations governing the Unity Engine’s physical simulation.

The currently-implemented solution in the most recent release of the algorithm is also admittedly the most naive. Since in most cases the aberrant Prouns display extremely fast movements followed by huge jumps (increases in average y coordinate) or explosions (all components fly far away in different directions), it is fairly trivial to check for a failed Proun configuration by tracking velocity and average height over its lifetime. Prouns are allotted 10 “flukes,” which allows normal Prouns to exhibit small bursts of movement (or “spasms”) without being flagged, but once a Proun is flagged it is given a fitness of -1 and removed from the scene. This effectively renders the Prouns sterile and we allow the normal Prouns to pass their genes onto the next generation.

While in a sense it is a greedy algorithm, only treating the symptoms instead of the cause, the filter works well for finding diamonds in the rough. Since we can save off Prouns that pass the aberrant behavior filter and maintain a high fitness in intermediate generations, we are left with a high number of Prouns that follow the aesthetic guidelines while doing reasonably well in the fitness function. We can also use the stabilityFitness measurement mentioned in section 4.2 to mutate Prouns with a low but nonzero number of flukes while keeping their topology and nodes constant in order to find a better control structure for a given body.
5 Results

Here we look at data collected during runs of the algorithm over approximately 200 generations, with 20 Prouns per generation given a maximum lifetime of 8000 simulation steps. At the end of each generation, the top 8 Prouns were kept to carry over to the next generation without mutating. See section 6.1 for a qualitative interpretation of the results.

**Figure 9:** Average fitness of all of the Prouns in a given generation (in blue) plotted with the best fitness among the Prouns in a given generation (in orange).
Figure 10: A scatter plot showing the relationship between Proun complexity and the number of flukes exhibited by the Proun in a given generation. Most Prouns are either completely stable or hit the fluke limit, while others either exhibit a few flukes or rapidly exhibit many flukes before being removed from the scene mid-generation. A trend line is drawn over the plot.

Figure 11: The total number of viable Prouns (with a fitness above average in the generation and exhibiting up to two flukes) collected in one run of 216 generations.
6 Discussion

6.1 Results

Starting with the fitness per generation, we do not see any overall improvement as the genetic algorithm progresses. This is not an appealing result for this type of algorithm. We would prefer the average fitness to improve over each generation, or at least settle into a local optimum, as “survival of the fittest” finds and preserves a certain elevated fitness. However, it seems that even though we are using elitism to keep high-fitness Prouns, some of those Prouns exhibit instability in future generations and are removed from the gene pool, keeping the gene pool from benefiting from the Proun’s high fitness over many generations. This could explain the brief peaks in “best fitness,” as a Proun is assigned high fitness because it travels a long distance, but it was only capable of traveling such a long distance because it was exhibiting aberrant behavior that was not violent or frequent enough to trigger the filter, but in the next generation the filter caught the behavior. See section 6.2 for potential improvements to the filter.

The trend line in the flukes vs complexity scatter plot (figure 10) weakly confirms the theory that increased complexity in the body of the Proun leads to an increase in the number of flukes in the Proun’s behavior. Here, complexity is measured as the multiplication of the number of nodes with the number of muscles in the Proun. Since many simple Prouns are also seen to exhibit a high number of flukes, the plot suggests that the Proun’s tendency to explode is not necessarily directly tied to the number of components that make up its body, but rather the way that those components are connected.

Finally, despite the fact that the genetic algorithm seems to perform poorly in the traditional sense for artificial evolution, we still see a steady collection of stable Prouns that are fit to be used in production. As a result, we still see a satisfactory end result.\(^6\) This is because while the fitness does not improve, all we need for the algorithm to be

\(^6\) See video titled “Proun Garden.”
successful is to find a diversity of stable configurations with a reasonable level of fitness. Since the algorithm does this reliably, we can say that the algorithm was ultimately successful in its original goal, regardless of its performance with respect to other, more traditional genetic algorithm implementations.

6.2 Algorithm Improvements
We have examined a unique genetic algorithm that borrows heavily from foundational work while contributing small yet important improvements that permit a modernized version of virtual creature evolution. By combining NEAT with Karl Sims’ strategies, and listening to errors that arise in corrupt genomes, we were able to build a new algorithm that generates virtual creatures which conform to the aesthetic guidelines of a professional digital artist without requiring any specific knowledge of the algorithm’s inner workings.

But the question arises: can we do better? The answer: certainly. Can the potential for destabilization be measured in a Proun before it is born, or even before it is conceived through a compatibility measurement between the Proun’s parents? Could we provide a control for managing the risk factor associated with trying complex, unseen topologies, where we can encourage the algorithm to produce more interesting results at the risk of losing more organisms to destabilisation?

While none of these questions has yet been satisfactorily answered through current experiments, the implementation outlined in this paper provides a solid foundation for investigating these questions. It could simply be a matter of combining the data currently being measured by the algorithm (stability, complexity, flukes, etc) with a record of seen topologies. Perhaps training a neural network to predict a stability metric given a genome would be useful for predicting future failures, and would extend elegantly to other output domains with aesthetic constraints of varying strictness and subtlety.

The current implementation performs a weighted coin toss on each gene in the parents’ genome. While it makes intuitive sense to do this, it is also possible that we are breaking apart successful sub-topologies which should be contributing to a more
successful, larger topology in the child. Since it seems that, as we concluded in section 6.1, the stability of a Proun is more dependant on the way in which its components are connected than the actual number of components, we are both encouraged to try more complicated configurations, but we also know that certain topologies (or sub-topologies) are more stable than others. If we can find a way to predict instability given a topology, we can use this information to build more stable Prouns that can share subsets of their topology in their offspring to begin constructively increasing the average fitness of the population across many generations.

Another shortcoming of this implementation is that the algorithm is only compatible with axis-aligned muscles and a small number of fixed joints per-Proun. It was nearly impossible to find a stable configuration that involved hinge joints like those seen in Sims’ original work. This seems to be another deep issue with the way that the Prouns are constructed. Right now there is no special consideration given to the way that topologies are programmatically constructed, and it is possible a dynamic programming approach would be more effective. Small topologies seen in simpler Prouns that are known to be stable could be more “trusted” than unseen topologies when mating genomes, and they would take precedence when being connected with other clusters of nodes and muscles to work in concert within a more complex (but stable) organism. If we are better at keeping Prouns stable as they grow in complexity, perhaps we can also begin to support more fragile mechanical components in the Proun’s body.

Finally, a subtler but equally important issue that should be addressed is the design of the fitness function. It is currently dependant on size, muscle density, distance traveled, and a rudimentary measure of stability. It is possible that the fitness function is simply incorrect for this type of creature. There is likely some less mathematically obvious measurement that would be more effective—some explicit representation of the degree to which the Prouns match the target aesthetics. Perhaps there is something about the Prouns’ orthogonal construction that makes travel challenging, and so Prouns that excel at it are those that violate the aesthetic constraints to do so.
Works Cited


