Evolving 3D Models Using Interactive Genetic Algorithms and L-Systems

Conference Paper · October 2017

DOI: 10.1007/978-3-319-69456-6_40

Shiqah Hadi
Institut Teknologi Brunei

Somnuk Phon-Amnuaisuk
Institut Teknologi Brunei

Evolving 3D Models Using Interactive Genetic Algorithms and L-Systems

Mariatul Kiptiah binti Ariffin(✉), Shiqah Hadi, and Somnuk Phon-Amnuaisuk

Media Informatics Special Interest Group, Centre for Innovative Engineering, Universiti Teknologi Brunei, Gadong, Brunei
mariatulit1990@gmail.com, shiqah@outlook.com, span.amnuaisuk@gmail.com

Abstract. The modeling of 3D objects is popularly obtained using a shell/boundary approach. This involves manipulating vertices and planes in a three-dimensional space using computers. Manually creating a 3D model in this way allows a designer full control over the creative processes but at the expense of long working hours. In this work, we explore the hybrid framework between *the Interactive Genetic Algorithm (IGA)* and the *L-system.* The L-system generates a 3D model from its production rules and the IGA evolves the 3D model by evolving the L-system's production rules. In this study, we investigate whether the approach can successfully steer the 3D model design using subjective preference feedback from users. We analyze and discuss the creative processes in the proposed hybrid system and present the models generated by our approach.

Keywords: L-system · Interactive Genetic Algorithm · 3D modeling · Genetic · Algorithm

1 Introduction

Modeling a 3D object manually is a laborious task, especially, when modelling complex objects with intricate shapes and design patterns. Altering a portion of the 3D model often requires changes that results in either adding or removing vertices and edges. These local changes propagate through the overall structure; therefore the overall model must be modified to suit new changes. Hence, we are interested in automating this design process using computers. We explore nature-inspired algorithms to automatically generate many 3D models and let a user interactively guide the design process using IGA.

In brief, each member in the GA population is an L-system object. The L-system has a set of production rules where it is employed to generate a 3D shape. The production rules can be viewed as a recipe for creating 3D models. Each production rule is a chromosome, in a GA context, where each chromosome is a finite length of string composed of terminal and non-terminal symbols. A 3D model can be rendered from the L-system chromosome [1]. Hence the modeling process is the iterative process of the following three main activities: (i) the rendering of 3D models where the system

S. Phon-Amnuaisuk et al. (Eds.): MIWAI 2017, LNAI 10607, pp. 485–493, 2017.
https://doi.org/10.1007/978-3-319-69456-6_40

interprets the chromosome strings and generates 3D models using the L-system, (ii) the evaluation of 3D models where the system accepts the user's interactive feedback, and (iii) the evolution of 3D models where GA evolves the L-system's production rules using its 2 reproduction schemes i.e., elitism, crossover and mutation. This process is repeated until the desired design is obtained. The paper is organized into five sections. The background of the genetic algorithm and the L-system is given in Sect. 2. Section 3 discusses the overall architecture and the key concepts of our method. Section 4 provides the illustration and discussion of the results. Lastly Sect. 5 provides the conclusions of the research and the future works.

2 Background

Applying genetic algorithm to a 3D modeling problem is challenging as the optimal solution is dependent on a user's subjective preferences. Translating a user's subjective preferences to computational constraints is difficult, if not impossible, as every human has his or her own preferences and it may not be explicitly quantified. This work resorts to the IGA framework which has been one of the popular approaches due to fitness functions' highly subjective nature, as those commonly observed in the creative domain such as music informatics [2, 3], user interface design [4], caricature generations [5] and product design [6–9].

Interactive Genetic Algorithm: In [6], the authors used IGA to evolve their preferred cola bottle shape design. According to the authors, IGA is chosen in this case because users have difficulty expressing their shape preferences as they are dependent on contextual and visual factors. The system allows users to flexibly and interactively express their preferences in a single evaluative quantity.

Depending on the domain, it may be natural to express the user's preferences using many criteria. In [10], the authors developed an IGA integrated generative design system using with multi-objective genetic algorithms. The IGA handles multiple objectives to optimize the qualitative and quantitative features in designing the artifact. The qualitative refers to the subjective human-based emotions which is a user-based fitness evaluation. The quantitative objectives refer to computable features such as the weight and size.

The L-System: Lindenmayer Systems (L-systems) describes a plant developmental structure using a set of production rules. These rules are expressed using strings of terminal and non-terminal symbols that can be interpreted as a 3D model. Hence, the modification of the L-systems' production rules allows for the exploration of the design search space. The authors in [11] describe a modeling system developed for the creation of three-dimensional animated models using the L-systems.

In [12], the L-systems are used to stimulate/for the stimulation of the evolution of artificial 2D plants. The genotype is represented by the mathematical formalism or the L-systems. The phenotypes are the branching structures resulted from the graphic interpretation of the genotypes. It provides IGA for a selection that allows users to control the simulated evolution through the phenotype. The overall results of the simulation have established that artificial evolution is a powerful tool for exploring a large and complex search space. But the experiment has only been tested on 2D graphics and employed the simplest type of L-systems (D0L-systems).

One of the old research methods [13], uses the 3Gmap L-system to model flowers. They combine the L-systems grammar writing with the interactive control of parameter settings. The models are created by operating the 3Gmap volumes. The flower model can also be modified interactively to allow users to create their intuitive model preferences. This modelling method also allows users to create the internal structure of the flower, giving more realistic results.

L-System also explores the 3D concept [14] for dynamic generation of agricultural crops. Here, the studies are more focused on generating crop heads using Fractal plant library. To manipulate the axiom and production of the 3D plant, it defines the type part and two angles; the minimum and maximum angle of the elements of parts. The parts consist of the element *stalk, leaf* and *bloom*, and provides individual information such as the radius, length and growth rate. The elements of stalk, leaf and bloom were created using polygon. E.g. the stalk can be represented using a green cylinder, leaves are represented using a green leaf-like rhombus and bloom is seven spheres that combines together to look like a flower.

3 Method

The proposed system is implemented in Blender. The L-systems and the GA components are developed in Python and are integrated in Blender as a software component interacting with the native Blender. Adopting dynamic language such as Python to L-systems [15] enhances the model by keeping the syntax simple, the code easy to execute and avoid compilation overhead. Models are also reusable and provide ways to build complex modular models. This enables easy access when running the simulation. The system has three main components as shown in Fig. 1: (i) the L-systems, (ii) the User-Evaluation, and (iii) the Genetic algorithm.

i. The L-systems component is a set of rewriting rules that translates a given string into a new string. It performs two main functions: (a) rewrite the string according to the rewrite rules, and (b) interpret the string as a 3D model and display the model. The L-system can be express as a tuple:

$$\text{L-system} = (\mathrm{V},\ \mathrm{S},\ \mathrm{P}) \tag{1}$$

V is a set of symbols containing both non-terminal NT = {A, G, T, C} and terminal symbols T = {F, Q, > , …, |}; S is a finite length start string $S \subseteq V^*$ and P is the production, here, only four production rules are defined for each model, $P = \{A \rightarrow V^*,\ G \rightarrow V^*,\ T \rightarrow V^*,\ C \rightarrow V^*\}1$. The table below summarizes some common terminal symbols and their semantics. A full detail of these terminal symbols and their corresponding semantics could be found in github (https://github.com/ento/blenderlsystem- addon) (Table 1).

The L-System program is implemented to Blender software as an addon. The parameters such as radius and positions are calculated using their own functions.

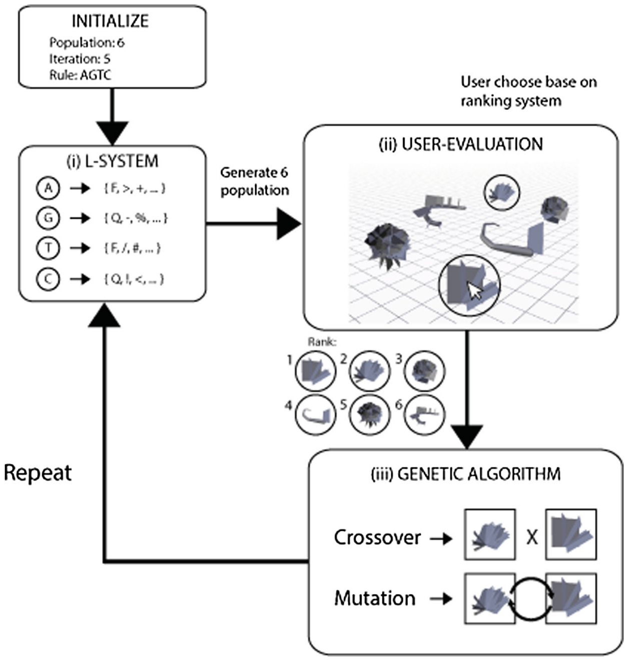

Fig. 1. The proposed framework, (i) L-system takes in the input and generate the models. (ii) User-evaluation lets user express their preferences based on models ranking approach. (iii) Genetic algorithm evolve the model using standard GA reproduction scheme: elitism, crossover and mutation operators.

Table 1. Summary of the terminal used in the system

Terminal symbols	Semantics
F	Create edge/branch segment
Q	Create quad or UV-mapped square/leaf
+, −	Rotate around the right axis (pitch)
/, \	Rotate around the up axis (yaw)
<, >	Rotate forward axis (roll)
[,]	Push or pop sack
&	Rotate random amount
!, @	Expand or shrink the size of a forward steps (branch segment or leaf)
#, %	Fatten or slink the mesh radius of a branch

The values are based on the user's input of the next component. With Blender's function, the 3D model is constructed using the instructions created by the L-System.

ii. The interactive user-evaluation component provides means for users to feedback their preferences back to the system. In each iteration, the L-system interprets and displays each chromosome as a 3D model. Users then exercise their preferences to select and rank these models. The selections are input manually via scripting console and these values will be calculated again through the L-System and GA.

iii. The GA takes the user's interactive preference feedback and uses it to guide its reproduction operations. Preferred models receive more opportunity for breeding. In our approach, GA performs standard elitism, crossover and mutation operations on the L-systems' rewrite rules. In other words, in each generation, the rewrite rules are evolved and the preferred models, i.e., preferred set of production rules are breeding.

4 Analysis and Discussion

We explore these concepts (finite generative rules and evolutionary strategy) to generate 3D models. Using Blender with L-system addons, we implement a hybrid IGA and L-system as a program component (IGAL) that communicates with the Lsystem. The IGAL performs all interactive GA functions (e.g., initialize population, evaluation, and reproduction) and communicates with the L-system (e.g., updating chromosome information). The IGAL starts with an initial population of L-system models. The number of models, the number of rewrite rule iterations, and other control parameters e.g., crossover rate, mutation rates, etc., can be determined by users.

The user then ranks their preferred models before repeating the generative processes until it creates a satisfied output. Figure 2 (top pane) shows representative examples of the generated 3D models. The models are randomly initialized in this case. The figure shows a gradual transformation towards the user's preferences. Comparing the first and its subsequent generation, the models appear to transform into new models using information from its parent's models (bottom pane).

Random evolution of 3D models may be preferred if one's design approach is in an organic style. Organic styles could produce interesting shapes and is a popular approach for evolving abstract shapes. The other alternative would be to start with a rough design and further evolve them. Our IGAL also allows users to define initial GA population with predefined 3D model.

Figure 3 shows representative examples of models evolved from different predefined shapes, here, a tree like structure. The models show different branching styles, balance and imbalance branching. As the generation progresses, we can observe that branching characteristics of different components are transferred from parents to their child model.

Though user's satisfaction is one of the main focus of the system, it does not completely adhere to user's idea of the model. Results are always mixed with one model being completely out of scope to their preference and the other might just be what user were looking for.

The idea that DNA is a recipe book for creating all organisms has been speculated in scientific communities long before we understand the structure of DNA. However, this topic is not well understood yet although we have learned a lot about the instructions to create proteins by DNA. This idea has also influenced many studies in computer generated models. Lindenmayer's L-system [16] and Wolfram's automata

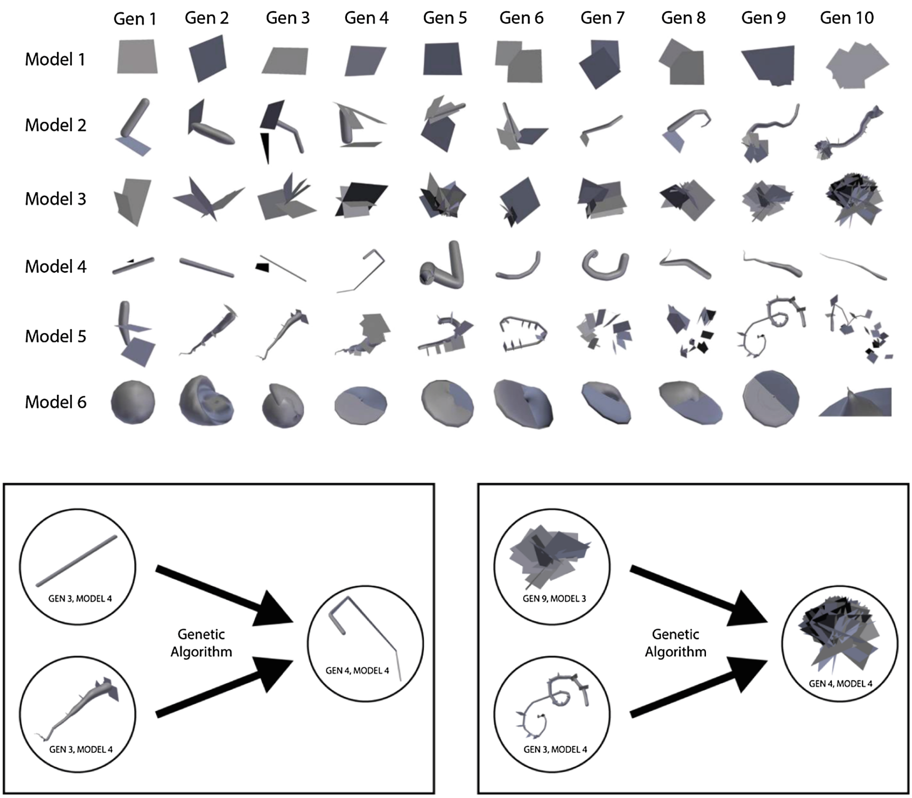

Fig. 2. Top pane – representative examples from random models initialization. Bottom pane - IGAL incorporates preferences of parents' 3D models to a new child model.

[17] explores the concepts that complex shapes can be generated from a simple set of rules. Dawkins further explores the concept of evolution. He implemented a program named *biomorph* which could create various forms which resemble trees, insects, birds (in a very abstracted shapes).

Challenges: It is observed that the proposed hybrid IGAL system could facilitate the 3D modelling process since the models are automatically evolved. However, there are many challenges in evolving the 3D models using IGAL. The following challenges are highlighted:

- Lacking specific/expressive feedback: In our implementation, preferences are expressed at a holistic level by ranking the models. The holistic approach is preferred since it is difficult to quantify different preferences. The lack of specific feedback makes it hard to control the direction of design. It may be important to allow hierarchical expression of preferences or multi-objective preferences.
- Lacking means to better steer the evolution: Lacking expressive feedback is one of the issue discussed earlier. Given a specific feedback, it will be fruitful if the information can be explicitly employed to influence the desired characteristics of

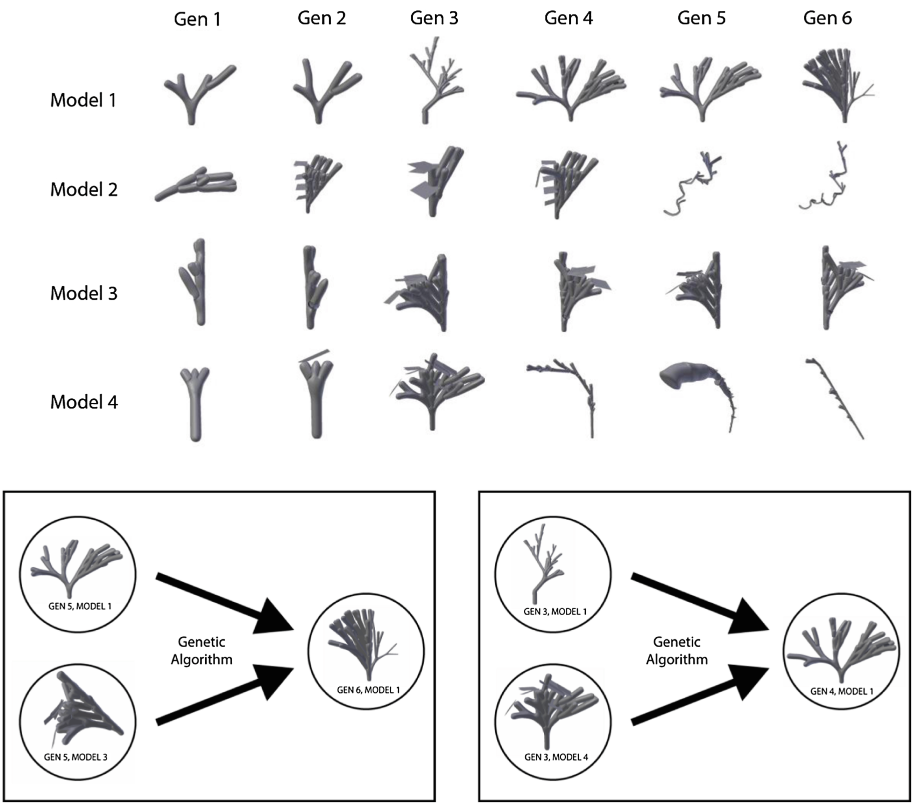

Fig. 3. Top pane – representative examples from predefined models initialization. Bottom pane - IGAL incorporates preferences of parents' 3D models to a new child model.

the model through specific knowledge intensive GA reproduction scheme. This requires some sort of control knowledge which may be expressed as a domain dependent knowledge source, or other machine learning tactics.

- Lacking semantic checking: During experiments, some of the generated models cannot be rendered by the L-system. We believe this is from non-well-formed combinations between the terminal and nonterminal symbols of the alphabets in the L-system. The current reproduction process (i.e., crossover and mutation) can only syntactically modify the production rules of the L-system. It does not perform semantic checking.

5 Conclusion and Future Work

This work explores ideas developed by Lindermayer, Wolfram and Dawkins. We have implemented a hybrid IGAL system that employs the interactive GA and the L-system to facilitate a 3D model design task. The approach has shown some potential and many challenges are visible to us during the course of this research. For future work, the system needs to be further developed. We have identified and discussed some challenges earlier and we wish to continue the research in those identified directions.

Acknowledgement. We would like to thank GSR office for financial support given to this research. We would also like to thank anonymous reviewers for their comments and suggestions.

References

1. Ochoa, G.: On genetic algorithms and Lindenmayer systems. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 335–344. Springer, Heidelberg (1998). doi:10.1007/BFb0056876
2. Koga, S., Fukumoto, M.: A creation of music-like melody by interactive genetic algorithm with user's intervention. In: Stephanidis, C. (ed.) HCI 2014. CCIS, vol. 434, pp. 523–527. Springer, Cham (2014). doi:10.1007/978-3-319-07857-1_92
3. Phon-Amnuaisuk, S., Wiggins, G.: The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system. In: Proceedings of the AISB 1999 Symposium on Musical Creativity, pp. 28–34. AISB, London (1999)
4. Kelly, J.C., Wakefield, G.H., Papalambros, P.Y.: Evidence for using interactive genetic algorithms in shape preference assessment. Int. J. Prod. Dev. **13**(2), 168–184 (2011)
5. Phon-Amnuaisuk, S.: Exploring particle-based caricature generations. In: Abd Manaf, A., Zeki, A., Zamani, M., Chuprat, S., El-Qawasmeh, E. (eds.) ICIEIS 2011. CCIS, vol. 252, pp. 37–46. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25453-6_4
6. Ma, L.: Research of product design based on improved genetic algorithm. Int. J. Hybrid Inf. Technol. **9**(6), 45–50 (2016)
7. Cluzel, F., Yannou, B., Dihlmann, M.: Evolutive design of car silhouettes using an interactive genetic algorithm (2010)
8. Dou, R., Zong, C., Li, M.: Application of an interactive genetic algorithm in the conceptual design of car console. Tianjin University (2014)
9. Kelly, J.C., Wakefield, G.H., Papalambros, P.Y.: Evidence for using interactive genetic algorithms in shape preference assessment. Int. J. Prod. Dev. **13**(2), 168–184 (2011)
10. Kielarova, S.W., Sansri, S.: Shape optimization in product design using interactive genetic algorithm integrated with multi-objective optimization. In: Sombattheera, C., Stolzenburg, F., Lin, F., Nayak, A. (eds.) MIWAI 2016. LNCS, vol. 10053, pp. 76–86. Springer, Cham (2016). doi:10.1007/978-3-319-49397-8_7
11. Viruchpintu, R., Khiripet, N.: Real-time 3D plant structure modeling by L-system with actual measurement parameters. National Electronics and Computer Technology Center, Bangkok (2005)
12. Boudon, F., Pradal, C., Cokelaer, T., Prusinkiewicz, P., Godin, C.: L-Py: an L-system simulation framework for modeling plant architecture development based on a dynamic language. Front. Plant Sci. **3**, 76 (2012)
13. Wyss, G.: Using L-systems for a dynamic generation of agricultural crops. BSc (Honours) thesis, University of Zurich (2013)
14. Petrenko, O., Terraz, O., Sbert, M., Ghazanfarpour, D.: Interactive flower modeling with 3Gmap L-systems. In: Proceedings of the 21st International Conference on Computer Graphics and Vision, pp. 20–24 (2011)

15. Boudon, F., Pradal, C., Cokelaer, T., Prusinkiewicz, P., Godin, C.: L-Py: an L-system simulation framework for modeling plant architecture development based on a dynamic language. Front. Plant Sci. **3** (2012)
16. McCormack, J.: Interactive evolution of L-system grammars for computer graphics modeling. In: Complex Systems: From Biology to Computation, pp. 118–130 (1993)
17. Wolfram, S.: Computation theory of cellular automata. Commun. Math. Phys. **96**(1), 15–57 (1984)