EG UK Computer Graphics & Visual Computing (2019)
G. K. L. Tam and J. C. Roberts (Editors)

Evaluating Models for Virtual Forestry Generation and Tree Placement in Games

Benjamin Williams[1], Panagiotis D. Ritsos[2], Christopher Headleand[1]

[1]University of Lincoln, School of Computer Science, Lincoln, UK
[2]Bangor University, School of Computer Science and Electronic Engineering, Bangor, UK

Abstract

A handful of approaches have been previously proposed to generate procedurally virtual forestry for virtual worlds and computer games, including plant growth models and point distribution methods. However, there has been no evaluation to date which assesses how effective these algorithms are at modelling real-world phenomena. In this paper we tackle this issue by evaluating three algorithms used in the generation of virtual forests – a randomly uniform point distribution method (control), a plant competition model, and an iterative random point distribution technique. Our results show that a plant competition model generated more believable content when viewed from an aerial perspective. We also found that a randomly uniform point distribution method produced forest visualisations which were rated highest in playability and photorealism, when viewed from a first-person perspective. Our results indicate that when it comes to believability, the relationship between viewing perspective and procedural generation algorithm is more important than previously thought.

CCS Concepts

• ***Software and its engineering*** → ***Interactive games;*** • ***Mathematics of computing*** → *Probabilistic algorithms;* • ***Computing methodologies*** → *Computer graphics;*

1. Introduction

Procedurally generating environments and landscape elements is becoming an increasingly important tool in games development. Generating content automatically is not only quicker than manual design methods, but also allows the possibility of creating perceivably infinite worlds. One particular area which is often overlooked is the procedural generation of forest and woodland bodies. Vegetation is an important detail in the design of a virtual environment, especially within the scope of natural landscapes. This is especially evident in modern video games, where virtual forests are frequently implemented as part of the in-game environment as scenery elements, but also to enhance game mechanics for, say, providing cover to players in first-person-shooter games.

Typically, virtual forests are designed through a manual or semi-automatic process in which a designer decides the distribution of individual trees throughout the scene. Not only is this process time-consuming, but the quality of the resulting scene is based on the subjective considerations of the designer. An alternative automatic approach is to position trees randomly throughout the environment according to a uniform distribution. However, real-world forests do not propagate in this way. The growth pattern of natural forests are instead governed by the development of an ecosystem over many years [CAB*00]. If the goal is to create scenes similar to real-world forests, then an approach which models this process may produce better results.

This paper presents an effort to undertake this challenge, by introducing a handful of generation techniques and placement strategies, followed by a user-based survey, evaluating each method in terms of perceived realism and playability. Furthermore, the attributes of the generated forestry (such as the density of the trees) are also studied to measure their impact on a player's perception of a generated forest. With this in mind, the hypotheses for this paper are as follows:

H1: A method which is an approximation of a real-life process (a bio-inspired approach) is perceived to generate more enjoyable and realistic content, over a stochastic method which uses randomness to distribute trees.
H2: The canopy coverage of each forest is a significant variable in the perceived playability and realism of it.

The structure of this paper is as follows: Section 2 reviews previous related work in procedural content generation, Section 3 presents three different approaches in procedural forest generation, Sections 4 and 5 discuss our pilot and main evaluations respectively, and Section 5 concludes, also presenting future work.

DOI: 10.2312/cgvc.20191259

2. Background

This section provides an overview of procedural content generation algorithms, and a review of their use in generating virtual foliage and flora communities.

In procedural content generation, content is generated stochastically via algorithms [TKSY11, YT11]. These methods have found success in a number of domains, including both research and commercial applications [HGS09, Vis10, PM01, Gam16, Boo09]. One area which procedural generation can be applied is the generation of virtual environments, such as dungeons [HHACT14], settlements [WH17], or in our case, plant ecosystems [BBHG02]. In this area, procedural techniques have been applied specifically to the generation of simulated vegetation. The majority of existing research into procedural vegetation focuses on generating individual items of vegetation, rather than an ecosystem built from individual plants. One of the most prominent methods for generating virtual trees procedurally, is through the use of Lindenmayer Systems (L-Systems) [Lin74]. L-Systems can be used to create fractal-like patterns, using re-writable grammars [Pru86]. These types of system are often used to generate the skeletal branches and stems of virtual trees [PBN*09, SRDT01, TFX*08, TZW*07]. In the work of Livny et al. [LYO*10], the authors even proposed an algorithm which reconstructed the skeletal system of a tree from a point-cloud through the use of L-Systems. The generation of other parts of a tree's structure, such as the bark, can also be generated procedurally. This was demonstrated by Dale et al. [DRHP14], in which the authors proposed a procedural technique for generating bark patterns, through a biomechanical physics model which emulated fractures in a tree's surface over time. Procedural methods have also been applied to generate other forms of vegetation, such as mushrooms [DVGG04] or lichens [DGA04].

An example of the earliest research in procedurally generating of systems of multiple plants is by Reeves and Blau [RB85], who explored the problem of how to generate virtual forests. A technique was developed which uses particle systems to approximate individual trees. The designer first defines a few parameters, such as the minimum distance between trees and the height-map of the terrain to place trees on. The algorithm then randomly distributes procedurally generated trees within the environment suited to the supplied parameters.

Another related class of algorithms are point distribution methods. There have been a number of papers which show their use in the procedural placement of objects, including trees and forestry [GLCC17, EVC*15]. A recent example of this is by Ecormier et al. [ENMGC19], in which a variance-aware disk-based distribution algorithm is presented. In particular, the authors highlight its usage in synthesising virtual forest scenes.

Other approaches, which consider plant competition models, have been developed. Plant competition models consider the simulation of each plant in an ecosystem, and interactions with its neighbours. Such an approach is presented by Bauer et al. [BBHG02] where the authors describe the *field-of-neighbourhood* (FON) model. The FON is a circular radius around each tree which determines the zone in which this tree competes with others in the community. If the FON of a tree overlaps with another tree's FON, then these trees are in competition with each other for resources. Otherwise, if there is no overlap between a tree's FON and another, then this tree is not in competition with any others. There are two competition models to consider if the FON of two or more plants overlaps: symmetric competition and asymmetric competition. Alsweis and Deussen [AD05] define these as:

- **Symmetric competition:** When considering the competition between two plants, resources are split evenly between the two. This infers that the two plants are of the same size, and pose an equal threat to one another:

$$I(a,b) = \frac{C(a,b)}{2}$$

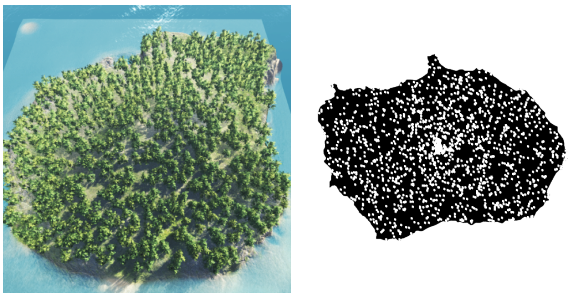
  where $C(a,b)$ yields the competition/FON-overlap between the two plants.
- **Asymmetric competition:** In the case of two plants, resources are split unevenly between the two, based on which FON is larger. This means that the tree with the smaller FON will be dominated by its competitor, resulting in no access to resources and its eventual death:

$$I(a,b) = \begin{cases} C(a,b) & \text{if } a_{\text{FON}} > b_{\text{FON}} \\ C(a,b) \text{ or } 0 & \text{if } a_{\text{FON}} = b_{\text{FON}} \\ 0 & \text{if } a_{\text{FON}} < b_{\text{FON}} \end{cases}$$

Alsweis and Deussen [AD05] use bio-inspired rules coupled with the FON model to generate plant communities through asymmetric competition. The development of a plant depends on a designer-supplied map which represents the amount of nutrition found throughout the terrain. Members of the simulated plant community reproduce by spreading their seed locally once they reach a certain size. The seed production of each tree also grows alongside its size – as it increases in size, it produces more seeds as a result. A 'mortality risk' is also introduced into the system, in which plants which fall below the average plant size are culled due to competition.

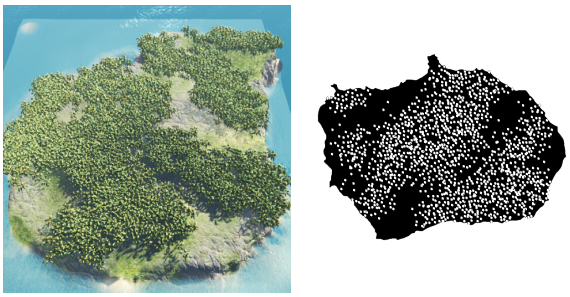
Lane and Prusinkiewicz [LP*02] use a similar approach to develop plant communities. In their method, a plant community is represented as a multiset L-System, in which individual strings of the L-System represent a tree. This multiset of strings is then added to or removed from to simulate growth within the forest. The authors also describe similar concepts, such as a radius around each tree in which it interacts with others (similar to the FON model) and domination of resources through asymmetric competition. To do this, the authors introduce the following three steps for each tree in the multiset:

- **Self-thinning:** A similar notion to asymmetric competition – plants which are in competition with larger ones are dominated, and are subsequently culled from the population.
- **Succession:** Trees grow over time, and have a random probability of dying at each step once they reach a certain age. This ensures that old trees are culled from the population.
- **Plant propagation:** Trees reproduce in a similar method proposed by Alsweis and Deussen [AD05], in which seeds are sown locally around the tree chosen for reproduction.



(a) (b)

Figure 1: a) *An example of a top-down virtual forest generated with the Naive algorithm, implemented in Unity 3D. b) An example in 2D.*



(a) (b)

Figure 2: a) *An example of a top-down forest image created using the Propagation algorithm, in a 3D environment.* b) *Another image generated using the same algorithm, but in a 2D environment. Both* a) *and* b) *were generated over a total of 13 iterations.*

Cordonnier et al. [CBC*16] draw attention to some scalability issues of FON-based competition models. In particular, the computational expense of FON models is moderate in smaller-scale simulations, but infeasible at larger scales. The authors introduce an approach to procedurally generate ecosystems with combined terrain generation. Instead of using a FON-based model, a non-competitive cell-based approach is used to simulate growth. In this approach, the landscape is subdivided into cells, and ecosystem events are generated at random in a given cell. Plant growth, death and germination are simulated based on plant viability. Plant viability is calculated by taking into account local temperature, soil moisture and sun exposure, amongst other factors.

3. Forest Generation Approaches

In this section we introduce three algorithms for the spatial distribution of trees within an environment. The first, the *Naive* algorithm, is provided as a baseline to evaluate the other methods against. This algorithm uniformly distributes trees randomly within the environment and is commonly used in games development. The second method is *Propagation*, based on a asymmetric plant competition technique, which implements the FON model discussed previously. This algorithm is a bio-inspired approach intended to approximate how natural forests grow over time. The third algorithm, the *Clustering* method is provided as an intermediary between the *Naive* and *Propagation* algorithms by using an iterative random distribution technique.

3.1. Method 1: Naive

The *Naive* method randomly distributes trees within a given area. The algorithm distributes trees by sampling a random (x, y) point in a uniform distribution, and places a tree at the sampled point. The algorithm used throughout this paper was adapted slightly to create forests at various densities. Instead of specifying a number of trees to spawn initially, a target density was specified and the algorithm ran until this target density was matched. Of all the methods described throughout this paper, the Naive method requires the least computational resources due to its simplicity.

3.2. Method 2: Propagation

The *Propagation* method takes its inspiration from the rules that govern how forests develop in nature. This method should not be considered a faithful reflection of a natural process, but rather a bio-inspired approximation. This method is based on the asymmetric plant competition approach described by Lane and Prusinkiewicz [LP*02]. We also similarly make use of a FON-based approach to approximate competition between trees. Furthermore, the three steps introduced by Lane and Prusinkiewicz within our algorithm are applied:

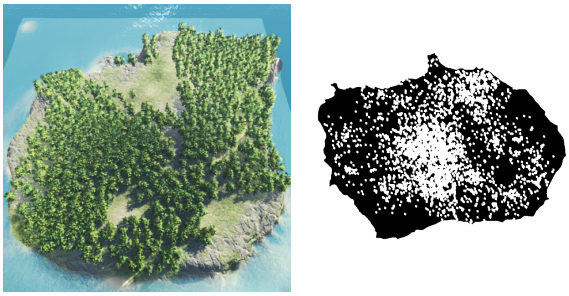
- **Succession**: In each simulation iteration, every tree ages (and grows) until it reaches a mature age. Once a tree reaches a certain age, it dies and is culled from the population.
- **Plant propagation**: Once trees have reached a mature age, they can reproduce by sowing seeds locally to their position.
- **Self-thinning**: If a tree is growing close to another tree, then the oldest (and largest) tree will outgrow the other, thereby killing it and culling it from the environment. This is an approximation of asymmetric plant competition.

In addition to these rules, the wind direction and wind magnitude are also simulated whilst generating the virtual forest. It is important to note that this is not an accurate simulation of nature, and various factors (such as evolutionary forces) are ignored. We accept this, and have simply taken inspiration from biology to try and generate something which is visually appropriate.

This method has the advantage of spacing the trees in a fairly regular manner. Due to the nature of the approach there should always be space between the trees, as if trees grow too close then the smaller tree will die. However, in comparison to the first two methods, this is computationally intensive as it requires many iterations before a forest can be fully generated. Whilst this would probably not be a problem for any standard gaming desktop, it may be an issue for mobile devices with limited computational power.

3.3. Method 3: Clustering

The *Clustering* method is an iterative random point distribution algorithm. In the first iteration, random points called 'spawn points' are sampled using a method similar to the Naive approach. In the



(a) (b)

Figure 3: a) *An example of top-down virtual forest generated with the Clustering algorithm, in a 3D environment.* b) *A similar forest generated with the same algorithm, but in a 2D environment.*

second iteration, positions are randomly selected within a radius of these spawn points to produce clusters of trees. Tree meshes are then placed in each of these final points to produce a forest. Likewise to the *Naive* method, the *Clustering* approach has the advantage of requiring very minimal resources, as the environment is not continuously updated and rules are not considered for each iteration of the forest's lifetime. This algorithm produces clustered distributions of trees, rather than an even and uniform distribution. Figures 3a and 3b show two examples of virtual forests generated with this algorithm, from an aerial perspective.

4. First Study: 2D Evaluation

An initial study was undertaken to evaluate whether the more complex approaches are preferred by players. The study consisted of an online survey where participants ranked images of top-down 2D representations of forests. The objective of this evaluation was to collect preference data regarding the visual forest representations.

For each question in the survey, participants were presented with three images of forests generated by each algorithm. Each image was randomly ordered on the screen, to reduce any selection bias between questions. The participant was then required to select one of these images which best matched the question criteria. The questions presented to each user throughout the survey evaluated two types of criteria. The first question was focused on the perceived realism of the environments. For these questions, the participant was asked to select two images (of the same three images) which they perceive to be the most and least realistic. The second criteria focused on the perceived suitability of the forest as an in-game environment. For this criteria, the participant were asked to imagine which environment they would (not) choose if they were to play a game based within this environment.

Both of these metrics are subjective to the observer. The first relies on them comparing the image to their perception/experience of what a forest should look like. The second by comparison explores their game-play preferences, assessing whether the environments perceived to be more (or less) believable are considered more (or less) interesting to play games within. Each participant was presented with five questions for each criteria, yielding a total of 20 individual questions. For each of the five questions, three new images were selected and presented to the participant.

4.1. 2D Study Results

The online survey was completed by 86 participants. Of these participants, 53.48% self-identified as female, with the remaining 46.52% as male. Furthermore, we also captured the general location of each participant, as the demographic featured participants from around the world.

The first and most compelling result found is the performance of the Naive distribution algorithm, which was comparatively rated higher than its competitors in terms of its perceived playability (see Figure 4). The Clustering method by comparison was rated as the method which produced the most forests perceived as most realistic. Figure 4 demonstrates that the Propagation distribution method was rated the lowest in terms of realism, but produced forests which were similar to the Clustering method in terms of playability. This same trend can also be seen for the questions which asked for the most unrealistic and unplayable environments (see Figure 5). For this category of questions, the Naive algorithm was similarly voted as the algorithm which produced the perceivably most realistic and playable environments. The Propagation algorithm however was rated as the most unrealistic and unplayable forest by a considerable margin.

Lastly, the number of ratings for each algorithm were used to provide a metric of performance, to show the overall quality of each algorithm. The metric used is calculated as $P_r = (R_r - R_{ur})$ and $P_i = (R_p - R_{up})$. R_r is the number of realistic ratings it received, R_{ur} is the number of unrealistic ratings, R_p is the number of playable ratings received and R_{up} is the number of unplayable ratings.

Figure 6 shows these two metrics plotted against each other, showing the overall performance of each algorithm. Interestingly, the performance of the Propagation algorithm was the poorest, producing the most unrealistic and unplayable environments. In contrast to this, the Clustering algorithm produced the most realistic environments, and the Naive algorithm yielded the most playable environments. The hypothesis was that the Propagation algorithm and application of a bio-inspired algorithm would produce more realistic and playable environments than the other two methods. However, the results show that the non-deterministic algorithms are rated higher in both categories. A further study is required to examine if this is the case under different conditions, and whether or not certain variables (such as forest density) yield similar results.

5. Second Study: 3D and Density Evaluation

A second study was conducted, in order to explore some of the questions raised by the first and to provide a more in-depth analysis of the reasoning behind selections. In this study the density of each virtual forest, along with the algorithm that produces it, were recorded and analysed. The participant also had the option of providing written feedback at every stage of each question.

As with the previous study, for each question asked, the survey presented the participant with three images to choose from. The participant would then choose the image which best suited the question that was asked. The questions were tailored in such a way to investigate whether the density or algorithm used in virtual forest propagation resulted in more playable or realistic selections. When

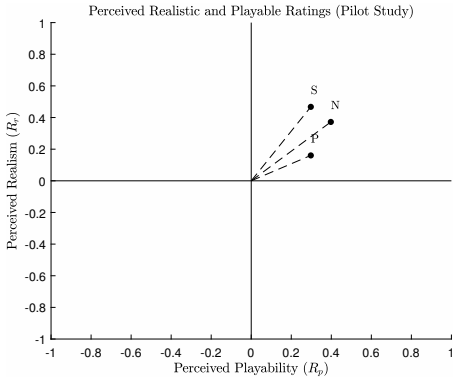


Figure 4: *The normalized number of responses from participants when asked to choose the most realistic and playable forest. The letters in this figure correspond to each algorithm used.*

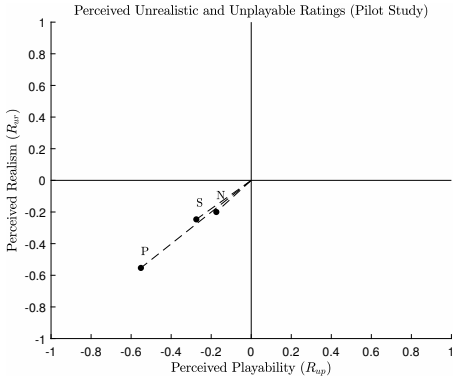


Figure 5: *The normalized number of responses from participants when asked to choose the most unrealistic and unplayable forest. The letters in this figure correspond to each algorithm used.*

selecting images to present to the participant, two independent variables were considered.

5.1. Algorithm Chosen

For these questions, the process started by first randomly selecting a forest density from the list of available options (Low, Medium or High). This density was then used to select three images for the participant, each of which was generated with a corresponding algorithm. For example if the randomly chosen density was 'Low', three low density forest images would be selected – one generated with the *Naive* algorithm, one with the *Clustering* algorithm, and another with the *Propagation* algorithm.

5.2. Forest Density

If the independent variable was forest density, then a similar process was followed, but showing varying forest densities generated with a single algorithm. To elaborate, an algorithm from the list of available options is randomly chosen (Naive, Clustering or Propagation). If for example, the randomly chosen algorithm was

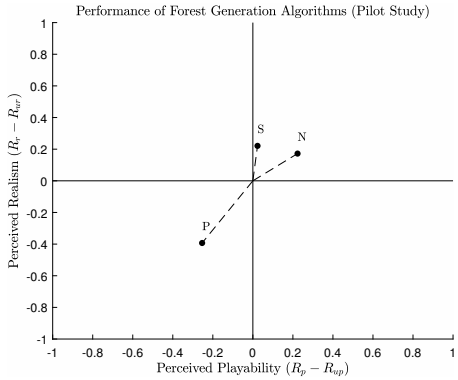


Figure 6: *The overall performance of each algorithm. Here the metrics used are the difference between positive and negative ratings.*

'Naive', then three forest images generated by the Naive algorithm would be displayed to the user – one with a low density, another with a medium density, and another with a high density.

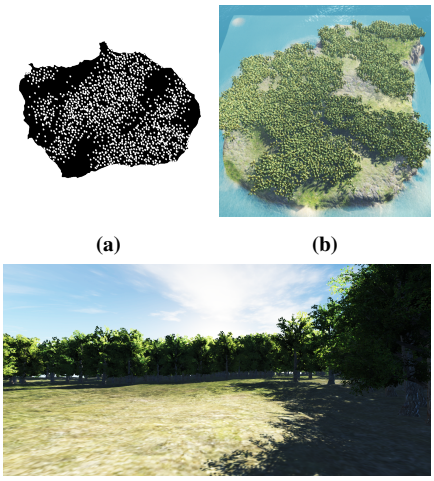
Once the three images were selected using these processes, the participant was then asked four questions about the selected images. These questions involved rating the forest images which best suited the question that was asked. These four questions were:

- 'Based on these images, which is the most realistic forest?'
- 'Based on these images, which is the least realistic forest?'
- 'If you were to play a game in one of these forests, which environment would you select to play within based on these top-down images?'
- 'If you were to play a game in one of these forests, which environment would you not select to play within based on these top-down images?'

5.3. Image Perspectives

Another limitation of the first study was that the images presented to each participant were from a single, top-down 2D perspective. This was addressed in the second study by introducing images which were rendered in 3D from two perspectives. Additionally, these images allowed further analyse if player perspective had an effect on a participant ratings. The first was a top-down perspective similar to the images from the pilot study, but rendered photo-realistically in 3D. The second used a first-person perspective situated within the forest. An example of the perspectives used in images can be seen in Figure 7.

These perspectives were also used in the question selection process. The same processes outlined earlier involving the isolation of forest density and the generation algorithm were used, but for every perspective. This means that eight questions were asked for each perspective, resulting in a total of 24 questions for the participant to complete. The study ran for three weeks in total, with 71 respondents. Of these 71 respondents, 77.46% were Male, 19.71% were Female, and 2.81% did not specify their gender. The following sections analyse responses given for each perspective.



(a) **(b)**

(c)

Figure 7: a) *An example of a top-down 2D perspective,* b) *An example of a top-down 3D perspective,* c) *An example of a first-person 3D perspective.*

6. Results

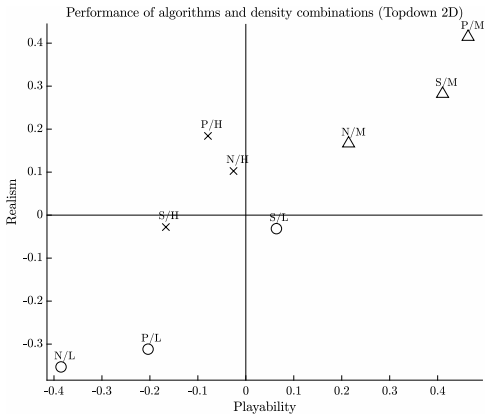
6.1. Top-down 2D Perspective

We plotted participant responses (Figure 8a), which measured the percentage a particular algorithm/density pairing (images generated with that density and algorithm) was chosen as playable versus the number of times is was chosen as realistic. The results show that images generated with the Propagation algorithm using a medium density scored higher in terms of both realism and perceived playability. An interesting result here is that the images generated with a medium density were rated similarly, and performed well in terms of both playability and realism. From this we can draw the conclusion that the most enjoyable forests for a top-down 2D perspective are generated with a medium density. It is also interesting to note that images of forests generated with a low density generally received a poor score. The exception however, are images generated with the Clustering algorithm using a low density, which was actually ranked higher in both realism and playability. Forests generated with a high density mostly scored well in terms of realism, but were rated low in terms of playability.

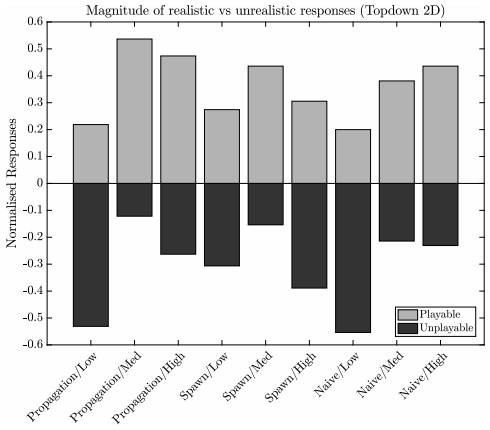
Figures 8b and 8c show the amount of responses provided for each particular combination of algorithm and density used to generate imagery. These figures also show in general, how many times a combination was rated negatively or positively. An interesting phenomenon regarding these is the amount of negative votes, which outweigh the number of positive ones. This means that participants who rated images generated with this perspective were more prone to select a negative rating rather than a positive one.

6.2. Top-Down 3D Perspective

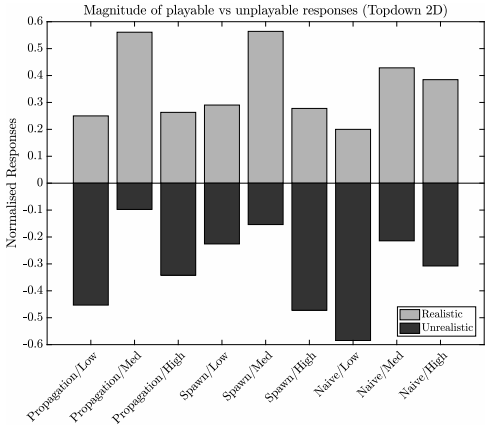
Through examination of Figure 9a, it can be seen that the results are similar to the ones found for the top-down 2D perspective (Figure 8a). Most notably, images generated with the Propagation algorithm using a medium density were again rated as the most realistic and playable environments. An interesting note however, is that images created using the Clustering algorithm have generally increased in both metrics, and are in fact some of the best performing



(a)

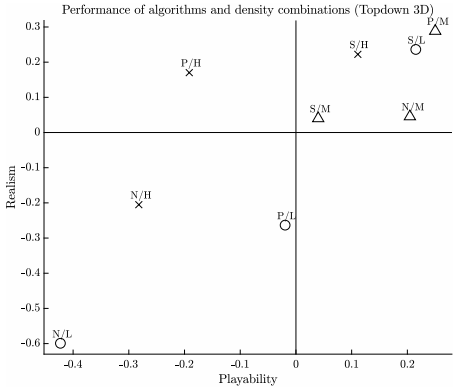


(b)

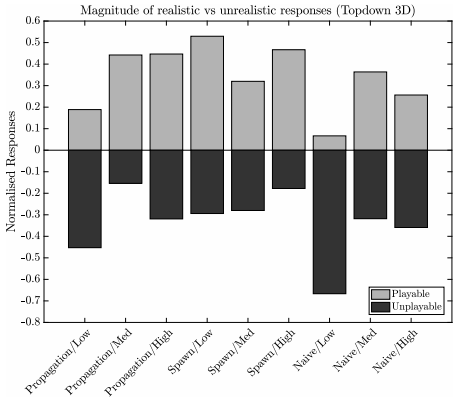


(c)

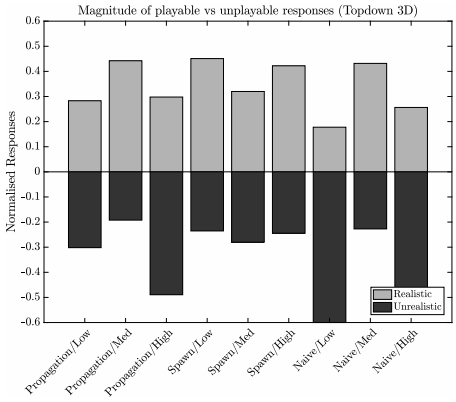
Figure 8: a) *Overall performance of all algorithm and densities for top-down 2D images, realistic rating vs playability rating,* b) *Magnitude of ratings for realistic/unrealistic responses and* c) *Magnitude of ratings for playable/unplayable responses.*



(a)



(b)



(c)

Figure 9: a) *Overall performance of all algorithm and densities for top-down 3D images, realistic rating vs playability rating,* b) *Magnitude of ratings for realistic/unrealistic responses and* c) *Magnitude of ratings for playable/unplayable responses.*

results. Figures 9b and 9c show the number of negative and positive ratings for generated images. These results are similar to the Top-down 2D perspective.

Images generated with the Propagation algorithm with a high density were rated well in terms of realism, but poorly in terms of playability. When compared to a lower density using the same algorithm, some intriguing results were found. Images generated with the Propagation algorithm but using a low density were rated high for playability, and low in realism - the opposite of the ratings when using a high density. The same algorithm is used to generate both types of images. The only difference between these two is the change in forest density. This contrast in terms of ratings leads us to believe that there may be a correlation between forest density and the perceived playability of an environment, when using this type of algorithm to generate an image of a virtual forest.

6.3. First-person 3D Perspective

The results were collated in the same manner as the previous sections. Figure 10a depicts rated realism and playability of images generated with each combination of algorithm and density. Interestingly, the results in this case differ from the results for the two other perspectives. The most compelling of these differences is that images generated using the Naive algorithm with either a medium or high density were rated the most realistic and playable environments. However, images generated with the Naive algorithm and a low density were rated lowest in terms of realism and playability.

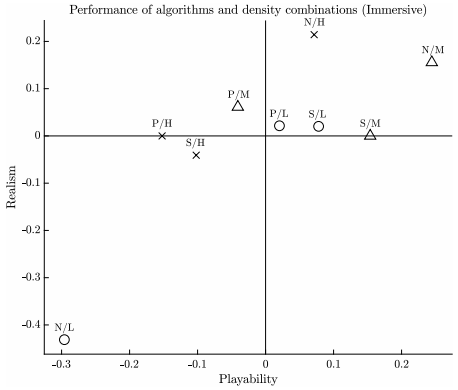
Comparing the results of using the Naive algorithm with medium and high densities further confirms the same correlation discovered in the previous section, in which the density used in the generation process affected its rated playability. In this case, the same relationship is shown – a higher density is rated as less playable than a medium density. This can also be seen in the same plot with the Propagation and Clustering algorithms, where a high density is rated less playable than a medium or low density. Furthermore, these results suggest that using a pseudo-random distribution strategy results in a more playable and realistic environment for players, at least, when viewing it from a first-person perspective. This has advantages over other methods, as it is computationally inexpensive in comparison, yet yields the most believable and playable environments for this perspective. Figures 10b and 10c show the number of negative and positive votes for images generated with each combination of algorithm and forest density.

7. Conclusion

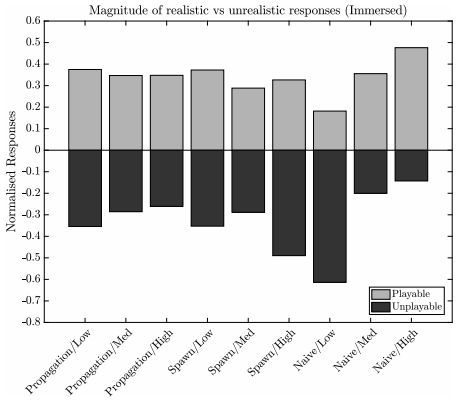
This paper presents a user study into virtual forests, using three different approaches of spatially distributing trees to approximate a plant community. These three approaches consisted of a random uniform distribution algorithm, a asymmetric plant competition model, and an iterative random distribution algorithm for creating clusters of trees.

Through this study, the results demonstrate that the asymmetric plant competition model (the *'Propagation'* algorithm) produces forests which were rated the highest in terms of playability and realism, for both a 2-dimensional and 3-dimensional aerial perspective. Interestingly however, a method which geometrically approximates asymmetric plant competition using pseudo-randomness to distribute trees (the *'Clustering'* algorithm) received similar ratings for the same perspective.

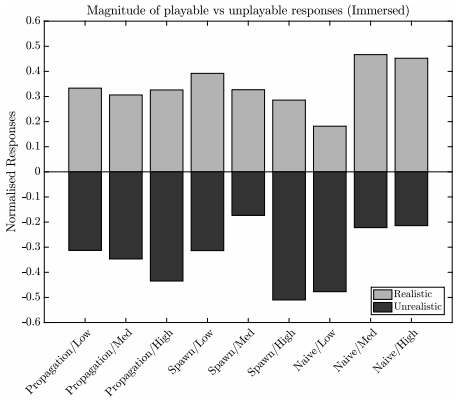
Another interesting result from this study is our findings when testing using virtual forest imagery from a first-person perspective. We found that the algorithms which score highly in the aerial perspective category were not scored as highly when viewed from the



(a)



(b)



(c)

Figure 10: a) *Overall performance of all algorithm and densities for first-person images, realistic rating vs playability rating,* b) *Magnitude of ratings for realistic/unrealistic responses and* c) *Magnitude of ratings for playable/unplayable responses.*

perspective of a player situated within the environment. Instead, we found that the control algorithm (pseudo-randomly distributing trees, the *Naive* approach) scored highly for both criteria when using this perspective.

We also found a relationship between the forest density used in images and their rated playability by participants. In particular, forests generated with a high density scored low in playability but highly in realism - whereas forests generated with a low density scored low in realism and high in playability.

From this, we can say that if the objective of the environment designer is realism and playability, they must consider the perspectives in which the forest is to be viewed when deciding on a procedural algorithm to generate it. If for example, the virtual forest is to be used within a game where the player is situated within the forest, the *Naive* approach could be used to create satisfying content while simultaneously conserving computational resources. On the other hand, if the virtual forest to be created is to be used as scenery from an aerial perspective, then employing the asymmetric plant competition approach may generate more satisfying content.

References

[AD05] ALSWEIS M., DEUSSEN O.: Modeling and visualization of symmetric and asymmetric plant competition. In *Eurographics* (2005), pp. 83–88. 2

[BBHG02] BAUER S., BERGER U., HILDENBRANDT H., GRIMM V.: Cyclic dynamics in simulated plant populations. *Proceedings of the Royal Society of London B: Biological Sciences 269*, 1508 (2002), 2443–2450. 2

[Boo09] BOOTH M.: The ai systems of left 4 dead. In *Keynote, Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE09)* (2009). 2

[CAB*00] CONDIT R., ASHTON P. S., BAKER P., BUNYAVEJCHEWIN S., GUNATILLEKE S., GUNATILLEKE N., HUBBELL S. P., FOSTER R. B., ITOH A., LAFRANKIE J. V., ET AL.: Spatial patterns in the distribution of tropical tree species. *Science 288*, 5470 (2000), 1414–1418. 1

[CBC*16] CORDONNIER G., BRAUN J., CANI M.-P., BENES B., GALIN E., PEYTAVIE A., GUÉRIN E.: Large scale terrain generation from tectonic uplift and fluvial erosion. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 165–175. 3

[DGA04] DESBENOIT B., GALIN E., AKKOUCHE S.: Simulating and modeling lichen growth. In *Computer Graphics Forum* (2004), vol. 23, Wiley Online Library, pp. 341–350. 2

[DRHP14] DALE H., RUNIONS A., HOBILL D., PRUSINKIEWICZ P.: Modelling biomechanics of bark patterning in grasstrees. *Annals of botany 114*, 4 (2014), 629–641. 2

[DVGG04] DESBENOIT B., VANDERHAEGHE D., GALIN E., GROSJEAN J.: Interactive modeling of mushrooms. 2

[ENMGC19] ECORMIER-NOCCA P., MEMARI P., GAIN J., CANI M.-P.: Accurate synthesis of multi-class disk distributions. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 157–168. 2

[EVC*15] EMILIEN A., VIMONT U., CANI M.-P., POULIN P., BENES B.: Worldbrush: Interactive example-based synthesis of procedural virtual worlds. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 106. 2

[Gam16] GAMES H.: "no man's sky", 2016. 2

[GLCC17] GAIN J., LONG H., CORDONNIER G., CANI M.-P.: Ecobrush: Interactive control of visually consistent large-scale ecosystems. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 63–73. 2

[HGS09] HASTINGS E. J., GUHA R. K., STANLEY K. O.: Automatic content generation in the galactic arms race video game. *IEEE Transactions on Computational Intelligence and AI in Games 1*, 4 (2009), 245–263. 2

[HHACT14] HEADLEAND C. J., HENSHALL G., AP CENYDD L., TEAHAN W. J.: *Randomised multiconnected environment generator*. Tech. rep., 2014. 2

[Lin74] LINDENMAYER A.: Adding continuous components to l-systems. In *L Systems*. Springer, 1974, pp. 53–68. 2

[LP*02] LANE B., PRUSINKIEWICZ P., ET AL.: Generating spatial distributions for multilevel models of plant communities. In *Graphics Interface* (2002), Citeseer, pp. 69–80. 2, 3

[LYO*10] LIVNY Y., YAN F., OLSON M., CHEN B., ZHANG H., EL-SANA J.: Automatic reconstruction of tree skeletal structures from point clouds. *ACM Transactions on Graphics (TOG) 29*, 6 (2010), 151. 2

[PBN*09] PRADAL C., BOUDON F., NOUGUIER C., CHOPARD J., GODIN C.: Plantgl: a python-based geometric library for 3d plant modelling at different scales. *Graphical models 71*, 1 (2009), 1–21. 2

[PM01] PARISH Y. I., MÜLLER P.: Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 301–308. 2

[Pru86] PRUSINKIEWICZ P.: Graphical applications of l-systems. In *Proceedings of graphics interface* (1986), vol. 86, pp. 247–253. 2

[RB85] REEVES W. T., BLAU R.: Approximate and probabilistic algorithms for shading and rendering structured particle systems. *SIGGRAPH Comput. Graph. 19*, 3 (July 1985), 313–322. URL: http://doi.acm.org/10.1145/325165.325250, doi:10.1145/325165.325250. 2

[SRDT01] SHLYAKHTER I., ROZENOER M., DORSEY J., TELLER S.: Reconstruction of plausible 3d tree models from instrumented photographs. 2

[TFX*08] TAN P., FANG T., XIAO J., ZHAO P., QUAN L.: Single image tree modeling. *ACM Transactions on Graphics (TOG) 27*, 5 (2008), 108. 2

[TKSY11] TOGELIUS J., KASTBJERG E., SCHEDL D., YANNAKAKIS G. N.: What is procedural content generation?: Mario on the borderline. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games* (2011), ACM, p. 3. 2

[TZW*07] TAN P., ZENG G., WANG J., KANG S. B., QUAN L.: Image-based tree modeling. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, p. 87. 2

[Vis10] VISUALIZATION I. D.: Inc.,"speedtree.", 2010. 2

[WH17] WILLIAMS B., HEADLEAND C. J.: A time-line approach for the generation of simulated settlements. In *2017 International Conference on Cyberworlds (CW)* (2017), IEEE, pp. 134–141. 2

[YT11] YANNAKAKIS G. N., TOGELIUS J.: Experience-driven procedural content generation. *IEEE Transactions on Affective Computing 2*, 3 (2011), 147–161. 2