Literature Review 1
Matthew DiBello
February 28, 2018

Introduction

Procedural modelling (PM) is the area of computer graphics concerned with generating three dimensional models. The paper *A Survey on Procedural Modelling for Virtual Worlds* (Smelik et. al.) took on the task of compiling the state of various areas of the PM world, and describing the many techniques currently employed. The first portion of this paper will be devoted to summarizing the findings of this survey paper.

Following this discussion, this paper will take a deeper look at one particular subsection of PM, specifically the work of Cordonnier et. al. in their paper *Authoring landscapes by combining ecosystem and terrain erosion simulation*. This will help to underscore the complexity and depth of the field of PM, and demonstrate one area of recent research.

A Survey of Procedural Modelling

The Benefits and Drawbacks of Procedural Modelling

PM has several advantages over manual modelling. The first of these is *data amplification*. In theory, good PM software should be able to transform a small amount of input data into a wide variety of procedurally-generated models. A second advantage is related, namely the *stochastic nature of PM*. Because much of PM is stochastic, it is very easy to create variations of existing models. A final advantage is *data compression*, or the idea that instead of storing data for a virtual world, if PM is used, a set of input information can be saved, and then later used for generation. The difference between storing all the information about a virtual world and storing inputs for later generation is significant, and it makes three dimensional models much easier (and cheaper) to store and distribute.

PM as it currently exists does have some drawbacks, however. The biggest of these is that the user of PM software has poor control. This is largely because current PM software is so complex and difficult to understand, that unless the user really understands the underlying mechanics of the software, he will likely be unable to use it as it was meant to be used. The result of this complexity is that much of the powerful software that has been created is virtually useless to anyone but the researchers and developers.

Terrain

One feature of virtual worlds that is commonly procedurally modelled is terrain. The most common method for generating terrain is by creating regular height fields, where each point in the XY plane has an associated height value for the terrain. The major drawback to this is that this method is useless for modelling overhangs and caves. Layered data structures, voxel data, and 3D meshes are some alternative methods that are more able to model more complex features of terrain.

Ongoing research areas for generating virtual terrain include erosion and weather simulations, and artificial intelligence. With erosion and weather simulations, some baseline terrain information is manipulated by functions that are meant to imitate the impact of erosion and weather on real-life terrain. The artificial intelligence approach is concerned with analyzing real-life terrain and generating virtual terrain that is similar.

Vegetation

For generating vegetation, there are three levels of detail to consider. At the most granular level, plant components can be individually generated. Alternatively, plants themselves could be generated as whole units. Finally, entire plant ecosystems can be generated and modelled. For vegetation generation, L-systems are frequently employed.

One major focus of this area of PM is generating vegetation that interacts with other aspects of the procedurally modelled world. This is important, since no plant exists in a vacuum, but rather its very existence is defined by its interactions with the environment at large. For this, environmental sensitive automata are employed, and plants are simulated as individual objects that compete for resources. This is an important principle that can also be applied to larger ecosystems and not just individual plants.

Water

The procedural generation of water bodies is not a very popular area of PM. The one exception to this is the modelling of rivers. For river generation, one of two main techniques is employed.

The first strategy is to construct the river first, then develop the rest of the world (height map) around the river. This method allows for a high amount of control over how river is constructed. Alternatively, the height map could be created first. After the height map is created, the river is begun at a high elevation, and the development of the river is left to the mechanics of physics. This method requires terrain that is moderately steep, and allows for virtually no control over the path of the river.

Buildings

Building generation is a well-developed area of PM, and most techniques use rewriting systems, typically either L-systems or shape grammars. Computer generated architecture (CGA) uses shape grammars to generate buildings. The main problem with CGA is that the shape grammars required for procedural generation are complicated and require a strong understanding to write.

For generating building interiors, there exist two main areas of interest: floor plan generation and furniture configuration. Shape grammars are often utilized for floor plan generation, though a subdivision technique is also employed. In the subdivision technique, sections of floors are organized, and more increasingly-granular aspects of the floor are planned out as the generation proceeds. Another method takes a user-defined graph of room connections and uses that to generate the floor plan. Furniture configuration is typically either based on a series of user specifications, or generated from a set of reference data.

Summary of Current Techniques

As was demonstrated by the above survey of various areas of PM, there are a few over-arching techniques. These are the stochastic, artificial intelligence, simulation and grammar-based methods. The major challenge across all of these methods is facilitating user control and creating intuitive PM software. A few features have been implemented, which help to increase user control and intuitiveness. One of these is a sketch-based approach, where the user can sketch out the skeleton of a design, at which point the software can take over and expand. Additionally, visual editors have been created, both for writing grammars and editing nodes and graphs. Techniques for inverse PM have been developed, which may eventually remove the need for much of the user control that is necessary today.

Before PM becomes widely used, several improvements must be made to the field. Firstly, the capacity must be developed for easy and intuitive editing and manipulation of PM. Methods across different areas of PM must be standardized to allow for the combination of the outputs from several distinct, independent, suites of software. Better support for the synthesis of manual and PM and the assimilation of PM into current development workflows are also critical.

Combining Ecosystem and Terrain Erosion Simulation

Overview of a New Framework

As technology advances, accurate models of virtual worlds are increasingly important. For this reason, Cordonnier et. al. developed a new framework for modelling the interaction between environmental phenomena and its effects on terrain and vegetation. With this framework, the user has a high level of control over environmental factors. A stochastic approach is utilized which is based on real data. The framework considers the life cycle of vegetation, as well as various environmental phenomena such as gravity, temperature, wind, fire and lightning. Events are iteratively applied over a duration of time.

The world is a grid of cells, which is comprised of two main sections, terrain materials and vegetation layers. The terrain materials include the bedrock layer, as well as the granular layers, which are broken rock, sand, and humus. Ground elevation, the slope of the terrain, soil moisture content, and average daily sun exposure are also tracked, as they are important factors. The vegetation layers are comprised of trees, shrubs, grass, and dead vegetation. The count, age, and height are tracked for trees and shrubs, whereas only density is considered for grass. The dead vegetation decomposes and becomes part of the humus layer. The total vegetation density is also recorded.

Simulation and Events

Instead of simulating numerous events simultaneously, a series of individual events is simulated. This improves the stability and dependability of the simulation, as well as its simplicity.

There are a few important rules for events. Firstly, events must travel through the grid from one cell to the next without looping or backtracking, though an event in a given cell can affect surrounding cells. Secondly, an event can only involve a limited amount of material. Finally, events are constrained to limited simulation time, which must be negligible when compared to a time step.

The timeline of an event is quite simple. First, a random cell is chosen as the origin of the event. Then a random event is selected. Next, the event is begun, and it continues until completion. Finally, the effects of the event are stored in the layers (moved materials, growth of vegetation, etc.).

There are two different categories of events. One such category is of the geomorphological events, of which there are several types of events. Rainfall and running water affect the terrain shape, vegetation growth and soil composition (bedrock is broken down and materials are moved). These events are influenced by the pattern of the terrain (which impacts the trajectory of the runoff) and the density of vegetation (as vegetation can absorb water). Temperature events can fracture bedrock due to thermal erosion and dramatic temperature changes. The amount of vegetation and other materials shielding the bedrock will influence the impact of temperature events. Lightning events can break bedrock down into broken rock, and destroy trees. Gravity-related events include rock-slides, sand-slides, and humus-slides, and are influenced by the friction of the material and the angle it's on. Finally, fire events can destroy vegetation, and are influenced by the wind (both its direction and intensity), as well as temperature, moisture levels and vegetation density.

The other category consists of the ecosystem events. For these events, there are several variables, including soil moisture, temperature, and sun exposure. These impact two factors, the vigor and the stress of vegetation. Vigor is the viability of the vegetation during seasons of growth. Stress is derived from historical vigor values, and is only impactful if it's negative. These two factors determine the germination, growth, and death of vegetation.

Analysis of the Framework

The framework appears to be very successful in several areas. First of all, the framework provides a tremendous amount of control to the user, which is a feature that is important in PM software. Additionally, the framework is fairly generic, in that it relies on a series of independently defined events that occur sequentially. This design allows future work with the framework to define new events and integrate them into the simulation. Finally, the quality of the simulation seems to be very good.

The framework does have its limitations. The computation time for the large number of events required for an accurate simulation is significant. Additionally, the simulation is done with a discrete grid. Even though this is important for keeping computation time down, it does decrease accuracy. Finally, validation of the simulation is challenging, as only qualitative validation is really at all possible.

Conclusion

The survey paper by Smelik et. al. provides an excellent overview of the field of procedural modelling. In the paper by Cordonnier et. al., we see a very specific application. Most critically, many of the principles laid out by Smelik et. al. as necessary for the widespread utilization of PM are applied to the created framework. The framework allows for a synthesis of manual and procedural modelling, and provides for a high level of user control. It is promising to see that researchers are already making great strides towards creating excellent PM software and paving the way for a future where PM is a staple of virtual world creation.

References

Guillaume Cordonnier, Eric Galin, James Gain, Bedrich Benes, Eric Guérin, Adrien Peytavie, and Marie-Paule Cani. 2017. Authoring landscapes by combining ecosystem and terrain erosion simulation. *ACM Trans. Graph.* 36, 4, Article 134 (July 2017), 12 pages. DOI: https://doi.org/10.1145/3072959.3073667

Ruben M. Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. 2014. A Survey on Procedural Modelling for Virtual Worlds. Comput. Graph. Forum 33, 6 (September 2014), 31-50. DOI: https://doi.org/10.1111/cgf.12276