Literature Review

Procedural Modeling of Cities Implemented As A Blender Plug-In

Kevin Mullane
g03m1642@campus.ru.ac.za

Abstract

The task of procedurally generating a three dimensional model of a city has been approached in a number of different ways. This literature review aims to review the approaches taken. Particular attention is paid to the methods used by Parish and Muller in [8] as it forms the underlying basis for this research project. Issues surrounding the type and style of virtual city generation and its relation to real world cities are discussed. The various methods of road network generation and building generation and texturing are discussed. Finally the conclusion is reached as to why the methodology provided by Parish and Muller is the one best suited to the requirements set forth by the "Text-To-Scene" system.

1 Introduction

The effective generation of a complex virtual city requires the combination of a number of individual issues which must be sufficiently resolved to allow for the modeling of that city. These issues range from the generation of a full transportation network to the generation of individual building structures which are both detailed and non-uniform. This research project uses as its base the methodologies employed by Parish and Muller in [8]. It is also however influenced by the various other approaches used in both virtual and real life city design. The prevalent characteristics of a generated virtual city are to a large degree defined by the overall goals of the generated model.

The requirements for virtual cities can range from large scale visually complex environments for use in the computer game or motion picture industries, to smaller scale but more realistically complex environments used for transport and population simulations. The aims of this research project dictate that its overall goal is to assist in the development of a larger overall project, namely the "Text-To-Scene" project which aims to generate complex three dimensional scenes based on descriptive input text such as novels. A greater focus is thus placed on the visually realistic characteristics of the generated virtual city as opposed to other characteristics such as population density or transport efficiency.

With the specific end goals of this type of virtual city generation in mind this review will aim to discuss firstly the major differences and similarities between real and virtual worlds including those attributes which are best suited for a visually complex city. Following this the two major aspects of city generation are reviewed, specifically the generation of a complex transportation network and then the generation of the individual structures within the city.

2 Real and Virtual Cities

The underlying similarities and differences between real and virtual cities need to be understood to allow for better prioritization of which aspects of real world cities need to be modeled. This will allow for the generation of a complex and visually realistic virtual city. This section will first examine the characteristics and influences upon real world city design as discussed by Rob Ingram [5] and Parish and Muller [8]. The relevance of these characteristics to virtual city generation and specifically the generation of a visually realistic city is then discussed. Finally the various approaches to virtual city generation are discussed including the methodologies of [3], [6] and [8].

2.1 Simulation and Fabrication

Two major approaches to the topic of virtual city generation have been explored, the first aims at city simulation and is based on data input which accurately reflects an existing real world city. The second is a more procedural approach which attempts to generate unique cities with little or no user input.

The methods used for city simulation depend on large amounts of accurate data being input into the system in various forms which may include aerial photographs or detailed statistical models of the city such as population densities. These techniques however rely on large amounts of input data and thus are not well suited to the overall goal of the “Text-To-Scene” project. In [4] the process of city generation from aerial photography is discussed and the end uses of a virtual city which is sufficiently complex and realistic as to mimic its real world counterpart for the process of simulation is discussed in [2]. These techniques however are not directly relevant to the outcomes of this research project and thus are not discussed further.

The process of city fabrication based on minimal input data however is discussed in [8], [6], [7] and [3]. The benefits of each of these approaches will be discussed further with specific reference to the requirements of the city generation system and the “Text-To-Scene” system. In order to successfully fabricate a virtual city however an understanding of the defining characteristics of a city as an entity are required as well as the relevance of each of those characteristics to the overall city structure.

2.2 Real World Cities

Ingram [5] discusses the required similarities between real world and virtual cities as falling under three broad categories. Firstly large scale design or the general layout of the city including the size and location of various areas within the city such as industrial and residential areas. Secondly the small scale design, this includes characteristics which are more directly related to the visual design and architecture of the various buildings and roads which constitute the city. Finally the concepts of governance and management are discussed, these include such overall layout decision as the zoning of particular areas to particular styles of structure such as housing in residential areas. All three of these categories play a significant role in the generation of a virtual world as they form the basic parameters which the system must incorporate in order to generate a realistic environment.

One of the advantages of a virtual city is that some of the restrictions which exist in terms of real world city design do not apply to the generation of a virtual world. These restrictions range from the existence of industrial pollution which prohibits the direct proximity of industrial areas with that of residential areas to the restriction that the city exist on a single plane which is governed by gravity. Theoretically a virtual city can have roads which run perpendicular to the landscape and allow for the generation of cities with structures based around various axis of orientation. As Ingram discusses however this will result in the loss of realism and a certain inability of the end user to be able to relate to the virtual city. Therefore any generation of a virtual city should follow some of the basic constraints which have governed the creation of real world cities.

Parish and Muller [8] cite some of the influences on real world city generation as being

> “historical, cultural, economic and social changes over time”

this results in the complex layouts and interactions which we experience in modern urban areas. This would suggest that the major contributing factor to a cities layout and design is in fact the various changes it experiences over a large period of time, older real world cities such as Paris which have existed from medieval times for examples are defined by the circular nature of their road network. This reflects the city's reliance on a circular wall for protection at some point in its history. Parish and Muller however do not suggest that a virtual world must take into account a huge range of variables over a set time period in order to generate a realistic city, they would suggest that rather the visual components of the city must be generated in such a manner as to accurately reflect the overall effects that these changes may have had. Specifically [8] discusses the use of road patterns a method of defining the overall nature and layout of the city. The raster or checker pattern they suggest most accurately reflects those newer cities which benefited from the intentional design of the layout of the road network. Cities such as New York which are characterized by evenly spaced rectangular city blocks use the raster pattern for road network generation. The radial or concentric pattern they suggest most accurately reflects such cities as Paris where the layout of the road network was based around the existence of a circular wall surrounding the city. A branching pattern may best reflect the natural progression of a smaller settlement or town where no concerted effort was made to plan the layout of the road network, these branches often reflect the shortest path between areas of higher population density or local environmental features such as rivers or valleys. By deciding on the overall pattern of the city layout during the generation process a visually realistic virtual city can be created without the need for complex influence simulations over time.

Christopher Alexander in [1] reviews the layout of two kinds of cities, specifically what he terms as natural cities or cities which have developed over a long period of time taking into account the effects of changing variables over time, and artificial cities which are planned in advance and develop over a relatively short period of time. This differentiation again highlights the effects that outside variables such as economic status and history can play on the layout and design of a city. Alexander suggests that we should envision cities as various collections of objects which interact with each other to form a much larger set which is the city itself. He suggests that each object within a city is a part of some set of objects and has a direct effect on those objects, these sets in turn intersect with each other so as to directly affect the existence of other sets. It is the collection of all of these sets of objects which can be most effectively defined as the city. Alexander then goes on to suggest that the components of this city set should not be seen as a tree structure where various sets and objects can exist on different levels of the tree and thus making the overall city set a sum of its constituent parts, but rather this set must be seen as having a semi lattice structure which better reflects the various relationships of each set with its constituent components and with other sets and their components. The resulting conclusion from this type of structure would suggest that any accurate simulation of a city would require the simulation of each of its constituent object and the accurate description of the relationship between each object and any other objects which it may influence.

2.3 Virtual Cities

The requirements of the city modeling system would suggest that some of the complexity raised by Alexander [1] may not be relevant to the type of virtual city which is being produced. Were the system to be aimed at the accurate simulation of population activities or traffic flow, then these more stringent requirements would have to be met. The three categories of city design as suggested by Ingram [5] are still relevant to the generation of visually accurate virtual cities. In the process of virtual city generation decisions must be made with regards to the overall layout and style of the city, road patterns such as those suggested by Parish and Muller[8] can be used to determine this. Small scale decision regarding the style of individual buildings and the type of roads used must also be made and must be in keeping with the large scale decisions. The realism of a virtual city would be degraded if these two categories were not similar in design, for example a city with large five lane highways but small wooden structures would not appear as realistic. The decision regarding governance and management however should act to ensure that the previous two categories are in keeping with each other, further restrictions on the cities layout can be enforced through its governance, restriction which would lead to the grouping of sky scrapers together in the commercial zones of the city and the grouping of small residential housing in the suburban outskirts. Whilst the stringent requirements of a real world city may be relaxed within the virtual realm, this ultimately results in a complete loss of realism, thus the more realistic that the various large scale, small scale and governance decisions are made, the more realistic and thus believable the resulting city will be. This level of complexity must however be sufficiently managed in order to create feasible virtual cities as reflected by Parish and Muller as opposed to the infeasible model suggested by Alexander.

2.4 City Generation Methodologies

Various different methodologies can be employed in order to create a virtual city each of which has its own defining features. Most of these methodologies however seem to agree on a clear distinction of the process into two separate parts, specifically the generation of a road or transport network and then the generation of city blocks and their constituent buildings. The various methodologies however have differing requirements in terms of input data, area planning and overall city requirements. Three methodologies will be discussed in further detail, specifically the methodologies employed by Mole [6], Greuter et al. [3] and Parish and Muller[8].

The first of these methodologies is that employed by Alexander Mole as a part of his creation of a Blender plug-in for the generation of complex city environments. This system requires that an existing model be used as input where building layouts are represented as simple geometric blocks within the scene. The methodology employed then aims to add varying levels of detail to that scene, specifically by texturing the blocks to more accurately reflect buildings, creating roadways between these buildings and then by setting up other smaller objects within the scene to handle street lighting and other effects. The overall aim of this methodology is focused at generating a smaller urban scene which consists of a fairly limited number of structures and a small regular road network. This is therefore not well suited to the requirements of the city generator and the "Text-To-Scene" system is likely to require cities of varying sizes and complexities. The techniques employed in [6] seem more focused on the process of procedural detailing than that of procedural modeling.

A second methodology for the generation of virtual cities is that employed in [3] which aims to create "Pseudo Infinite" virtual cities. This is achieved by generating a large square grid upon which various building models are placed. This methodology does not take into account any form of city layout or road planning, but rather relies on the large range of building design to give the impression of a larger city. Each individual bailing is generated by continuously scaling and rotating a building footprint upwards to create a random overall building shape. This process is particularly well suited to the creation of taller sky scraper types of buildings. The overall methodology employed does not allow for the open ended requirements of the "Text-To-Scene" system. The implementation of a road system with this methodology is to generate a perfectly square grid of roads which is so large as to make it unlikely that the user would ever reach the edge of the city. This limits the usefulness of the generated city as it cannot be effectively incorporated into any existing scene and thus could not be used for the "Text-To-Scene" system.

For the purposes of this research project, the methodologies employed by Parish and Muller are the most relevant and thus form the basis for the methods used. This methodology is the one that is most in keeping with the three major criteria discussed in [5], it relies on the input of image maps to determine any natural barriers to city development such as rivers or mountains. This input however can be easily replaced by information drawn directly from the scene into which the system will be generating the city. Once the legal and illegal areas for development have been determined on the map, the system uses an complex self-sensitive L-System to generate a road map over the landscape. This road map is then used as a a guide to divide the reaming spaces up into lots through a simple subdivision algorithm. These lots are then populated with simple buildings using another L-System which generates strings to represent the geometry and logical operations which will be used to represent each building. Finally that geometry is generated resulting in a full complex model of a virtual city. This methodology appears to represent the most versatile and flexible system for procedural city generation.

3 Road Network Generation

The process of procedurally generating a virtual city that conforms to the requirements laid out above begins with the generation of a complex road network. This road network must conform as much as possible to a real world road network and thus is controlled to a large degree by outside factors such as the underlying terrain and the overall style of the city being generated. Various methods for road construction are suggested in the relevant literature include the simple square grid system of roads used in [3] and the process of defining roads after the creation of the structures as implemented in [6]. The most common method and the most relevant to methodology which this research projects aims to follow is that of template or pattern based generation of road networks and suggested in [8] and [9].

3.1 Template based road generation

In conforming to the criteria suggested in [5] which suggests that a generated city should have large scale design decisions and small scale design decisions, the process of template based road design as suggested in both [8] and [9] allows for the road network to be generated dependent on an overall city style. One of the main features of both of these methods is that they differentiate substantially between highways and side streets, both suggest that patterns or templates be used to generate a large network of highways and then the gaps between these highways be filled with a simple straightforward grid pattern of side streets.

Parish and Muller discuss the use of population densities to determine the path of each segment of highway. This relies on the premise that highway roads will naturally evolve to connect neighboring areas of high population density. This process gives each successive highway segment a general direction and length, the actual placement of the highway however then relies on the conformation of that highway segment to some overarching road pattern which controls the layout of the road network. These patterns can be basic which suggests that there is no superimposed pattern and that the highway segments simply branch naturally from one point of population density to the other. The rectangular or raster pattern defines each successive segment with a specific angle and a maximum length, by setting the angle to be either 90 or 180 degrees, and by keeping the road length constant this pattern can be used to generate a perfectly square grid pattern. The radial pattern adjusts highway segments to follow circular paths around a central point which can be defined randomly or through population density. A final road pattern that is discussed in [8] is the San Francisco rule where each highway segment follows the path of least elevation and thus results in a road which simply follows the contour of the underlying terrain. These highway segments are then connected by smaller side streets which are short and follow the steepest possible path from each point to facilitate the connection of the highway segments.

Sun, Baciu, Yu and Green in [9] suggest a system which is very similar to the one suggested by Parish and Muller. Their system is also based on an overarching city pattern which defines each segment of highway, they however only discuss three types of template from which to design a city, the first is a population based template which follows much the same rules and the basic branching pattern proposed by Parish and Muller, the second is a raster and radial based template which is again very similar to the Parish and Muller suggestions and finally they propose a mixed template which combines the effects of both the population based and raster templates to generate mode complex and realistic city scenes.

3.2 Generation Methods

The process for initial road segment generation in both [8] and [9] uses the concept of beginning at a point, usually the end point of the previous road segment and then generating a spectrum of rays emanating out from that point. Each of the end points of these rays are then checked against the chosen road pattern and the natural limitations of the terrain in order to determine the best possible choice. Once the best possible ray has been chosen, it is converted into a road segment and the process is repeated. In Parish and Muller the possible rays are passed through an extended L-System or road generation grammar which determines which variables are most relevant to each road segment and what their possible ranges are. It may decide that the road cannot rise at a elevation about 20 degrees and thus will remove all possible rays where the end point results in a road segment with such an elevation. It depending on the chosen pattern may aim for the point which has the highest population density. The process implemented in [9] whilst not specifically called a construction grammar or an L-System appears to perform much the same form of calculations in order to determine the initial position of the next road segment.

3.3 Local Constraints

The process suggested by Parish and Muller differs from that suggested by Sun et. al. in its application of local constraints. Once the L-System has determined the best possible road segment from the collection of rays generated, the road segment must be checked against a range of local constraints to determine if it is in fact a valid road segment. Local constraints include such checks as the intersection of a road segment with another road segment. If two roads intersect and the resulting road segment is too short or the angle between the roads is two small, the road fails the local constraints check. A further check is to determine if there exists any other road segments or intersections within a set proximity from the road ending. This is used to connect the ends of roads back into the road network and allow for a continuous flow of road segments. If a road segment fails to meet the requirements of the local constraints, then the system attempts to find a suitable location within the roads immediate vicinity, this helps to overcome the problem of roads continuing into illegal areas on the landscape, if a road segment passes over an illegal area, the system will shorten the road until it no longer crosses that boundary and thus becomes a legal road. If no legal road segment can be generated in the immediate facility then the road segment is simply ignored and the process of segment generation continues. The conclusion of this method is the generation of a complex highway network that follows not only the overall constraints of the road pattern, but also generates safe road segments which do not clash with any of the natural constraints associated with the landscape.

3.4 Shape Modification

Road segments are rarely ever straight lines and thus some provision must be made within the generated road network for curved road segments. In [9] Sun, Baciu, Yu and Green suggest a method of shape modification which converts a straight segment of roadway into a curved and twisting segment dependent upon the underlying terrain. The process involves recursively breaking each road segment into two equal halves and then adjusting the midpoint according to the elevation and distance from the endpoints. The system attempts to minimize changes in elevation for each successive midpoint, this therefore results in the generation of road segments which curve naturally with the contour lines of the underlying terrain. The more times this process is applied to the road segment the more curved the road will become. This process of recursive shape modification can be used to alter the road segments according to any number of factors dependent on the road template used.

4 Building Generation

Once the process of road generation is complete the internal regions formed by the various road segments can be converted into city blocks and populated with buildings. There are a wide range of methods for building generation ranging from simple rectangular blocks with procedural textures to complex shape generation grammars which produce various building shapes. A texture is then added to the surface of the buildings geometry to enhance the realism of the scene. For simple rectangular blocks complex procedural texture as are required which specifically map such objects as windows and doors onto the blocks in order to give the impression of many different types of buildings. The more complex the initial geometry the simpler the added texture can afford to be as the buildings are no longer relying solely on that texture for their unique nature.

4.1 Division of Lots

The first task associated with procedurally modeling buildings is that of lot division. The regions between the road segments generated for the scene must be populated with buildings. In [8] city blocks are created by scaling downwards from the road edges and intersections. These blocks are then broken up into smaller rectangular shapes through the use of a simple subdivision algorithm. Any resulting rectangle that does not face directly onto a street is deleted to reduce the quantity of work which the system must perform and the remaining rectangles are populated with building structures.

4.2 Geometry

A combination of complex geometry and complex textures can be used to generate a large variety of realistic building models for use in the virtual city. The basis for this is the underlying geometry of the buildings. Parish and Muller [8]use a combination of scale and rotation transformations upon a simple geometric object to derive complex building models. Specifically a shape is created which represents the footprint of the building, this shape is then extruded upward and becomes the base of the building. The shape is then copied and transformed upward on the building to rest upon the previous shape, a shape grammar is then applied to the shape, this adds a number of scale and rotation operations to the shape within a set of parameters defined by the grammar. The resultant shape is smaller than the previous shape and sits above it on the building. This process is repeated with different scale and rotation transformations being applied each time as the building begins to grow upwards. The result is a geometrically complex building shape upon which various textures can be applied.

An extension to this method of building generation is provided in [3] where the same process of footprint extraction is followed, with this method however the shape is constructed in the opposite direction. The initial polygon used forms the top of the building shape and as further steps for transformation are applied to each successive shape, the original shape is extended downwards. Thus once the process of building generation is complete, the initial;l shape still exists at the top of the building, but it also extends down through the building to the floor. At each successive step a new simple polygon is generated which is in no way linked to the previous shape. This polygon is then shifted slightly off-center and rotated. The new shape is then extruded downwards by a variable amount and the process repeats itself. This method of generating the building geometry from the top down with various unrelated polygon shapes allows for even more complex and unique building shapes. This method however applies mainly to taller buildings and does not appear suitable for house type structures.

Another extension to this system is provided in [7] where the bottom up approach is used, but it is combined with the idea of multiple polygon shapes. Thus a complex building footprint is generated through the combination of various simple polygons and each polygon is extruded up to a unique height generating complex building geometries. [7] also adds to this method my providing for roof surfaces which are generated at the top of each polygon shape to give the buildings unique sloping roofs as opposed to the simple flat top of the extruded shape.

4.3 Procedural Texturing

The true complexity of the individual buildings generated for the virtual city are derived from procedural texturing techniques. [8] and [10] both describe systems which use shape grammars to translate the co-ordinates of a large sample texture to the specific co-ordinates of the relevant face of the building shape. By successfully implementing these shape grammars, a large sample texture which includes specific regions to represent doors, windows and other unique areas, can be used to map the door region of the grammar to the from bottom face of the building geometry. The results of this is the ability to turn fairly non-descript shapes which represent the individual buildings into complex unique shapes each with their own combinations of windows and doors. Wonka et al. [10] implement a more complex shape grammar then Parish and Muller and are able to map specific areas of the texture to specific regions of the shape, not only dependent upon the building shape, but also dependent on the neighboring textures. This allows for such combinations as windows and separate window sills. This allows for a relatively few sample textures to be used whilst allowing for a huge number of unique building combinations.

5 Conclusion

The task of generating a procedural model of a virtual city which conforms to the requirements of the “Text-To-Scene” system provides many challenges and complex constraints. If some of the broader characteristics of real world city design are successfully implemented as is suggested by the methodologies reflected in [8] then the challenges can be overcome. Specifically a city can be fabricated using the minimal input or even random choice of a road pattern and building style. From this a complex network of highways and side streets can be generated which conforms top both the constraints of the landscape and the constraints of the city itself and the regions between these roads can be converted into a wide array of unique and complex building structures. The results of this process will be the procedural generation of a complex and visually detailed city.

References

[1] Christopher Alexander. A city is not a tree. *Architectural Forum*, 122(1):58–62, April 1965.

[2] Luiz Gonzaga da Silveira and Soraia Raupp Musse. Real-time generation of populated virtual cities. In *VRST ’06: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 155–164, New York, NY, USA, 2006. ACM Press.

[3] Stefan Greuter, Jeremy Parker, Nigel Stewart, and Geoff Leach. Real-time procedural generation of 'pseudo infinite' cities. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 87–94. ACM Press, 2003.

[4] Olof Henricsson, Andre Sreilein, and Armin Gruen. Automated 3-d reconstruction of buildings and visualization of city models, 1996.

[5] Rob Ingram. Building virtual worlds: A city planning perspective. In *Proceedings of the Fourth UK VR-SIG Conference*, Brunel University, November 1997.

[6] Alex Mole. Procedural city detailing. Honours Thesis, 2004.

[7] Pascal Muller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. In *SIGGRAPH '06: International Conference on Computer Graphics and Interactive Techniques*, pages 614–623, New York, NY, USA, 2006. ACM Press.

[8] Yoav I. H. Parish and Pascal Muller. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308. ACM Press, 2001.

[9] Jing Sun, Xiaobo Yu, George Baciu, and Mark Green. Template-based generation of road networks for virtual city modeling. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 33–40. ACM Press, 2002.

[10] Peter Wonka, Michael Wimmer, Francois Sillion, and William Ribarsky. Instant architecture. *ACM Transactions on Graphics*, 22(3):669–677, 2003.