

Rapid SmartCode Modeling

Procedural Modeling for Urban Environments

Yoshihiro Kobayashi, PhD.

Arizona State University, Japan.

dr.yoshihiro.kobayashi@gmail.com.

Michael McDearmon

Arizona State University, USA.

Michael.mcdearmon@asu.edu

Abstract. This paper presents an image-driven method to rapidly generate 3D digital urban models for presentation in a real-time Virtual Reality (VR) environment. Several computational tools and methods are developed in order to give the end user the ability to transform a single 2D image into a realistically textured and rendered a 3D urban model. Using this method, city models based on SmartCode, a template for planning and urban design, are generated and presented in a VR visualization tool. The methods to generate VR city models including inputs, process, output, and pipeline are explained. One generated 3D city model is demonstrated in the results section.

Keywords. 3D city modeling, urban design, virtual reality, SmartCode, MaxScript

Introduction

Presently, 3D digital urban models are abundant in GIS, Google Earth, 3D games, architectural renderings, and urban design visualizations. However, based on the time and development costs of producing these models, they are often not able to be created for master planning discussions and are instead used in visualizing a final design for advertisement. A very similar problem exists in urban design. It is currently difficult for stake holders and decision makers to capture a clear vision of output design in the initial planning phase. In order to solve the problem, form-based codes like SmartCode for visualizing output designs have rapidly been developed and used more practically (SmartCode, 2009). The motivation of this research is to implement computational design tools that generate 3D VR city/urban models by combining procedural modeling techniques with SmartCode.

In addition to providing a powerful tool for master planning discussions, the design tools developed in this project significantly decrease the labor-intensive process of 3D city modeling, thereby having the potential for application in 3D animation, Hollywood films using 3D CG, consulting companies using form-based codes, video games, and television.

Several rapid and procedural modeling programs for urban environments already exist and served as inspirations for this research. CityEngine is a popular procedural modeling tool designed to reduce the production costs associated with creating 3D city models (Procedural Inc., 2009). This standalone software package includes numerous tools for generating street maps, 3D buildings, and 3D street models. Procedural modeling techniques were also employed in the creation of Rama, an innovative animation featured at Siggraph 2006. The animation features a re-make of New York City created entirely by ANSI C programming (Bruneton, 2005). A number of other studies conducted by Yoav I. H. Parish, Peter Wonka, and George Kelly all delve into various methods of rapidly generating 3D urban environments.

Though the computational tools developed for this project are inspired from existing software and techniques, they innovate in several key areas that the existing technology does not. CityEngine necessitates a basic understanding of scripting in order to generate 3D buildings. Additionally, both CityEngine and Rama were implemented in their own platform, so it is difficult for design professionals, students, and others to learn and test these tools. In

this research, however, procedural modeling tools are implemented as a plug-in for the industry standard modeling package 3D Studio Max and require no scripting, so users with various levels of familiarity with the software can learn the framework more quickly (Autodesk, 2009).

Another key innovation with this research is that our procedural modeling tools for 3D cities are based on the planning and zoning guidelines known as SmartCode. The code is described as “a unified development ordinance that incorporates the transect-based planning principles of Smart Growth and New Urbanism” (Duany, 2008). The notion of the transect, a pedestrian scale cross-section of a city containing six unique zones from natural to urban (see fig. 1), is both a guiding principle of SmartCode and an impetus of this procedural modeling framework.

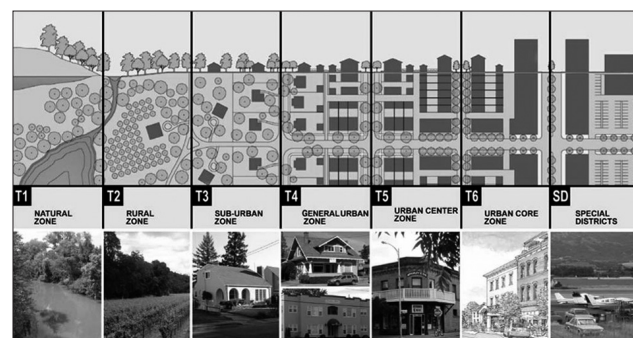


Figure 1. SmartCode Transect (Credit: Duany Plater-Zyberk & Co.)

Methods

The procedural modeling application developed in this research, called City Design, is a plug-in for 3D Studio Max that sequentially models, textures, renders, and exports a city model for use in virtual reality (VR) applications, gaming environments, movies, and other platforms. The framework of this procedural modeling technique is outlined in Figure 2.

The process begins by the user creating four simple input files that specify the shape, size, terrain, and zoning of the generated city model. The City Design plug-in then automatically generates a 3D urban environment based on these four inputs including two drawing files and two image files:

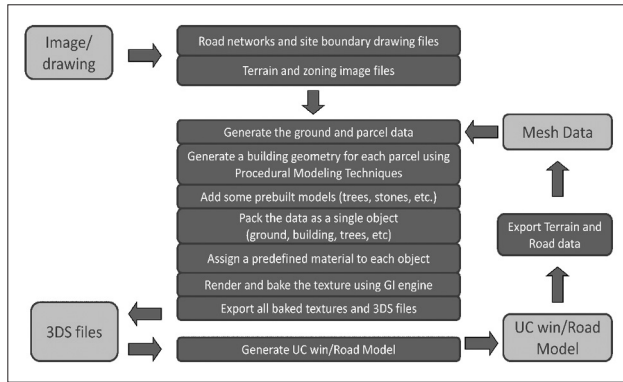


Figure 2. Outline of procedural modeling framework.

Drawing inputs:

Two simple drawing files, created using Adobe Illustrator or similar software, will be used for defining the shape and size of the generated city model. One file is used to specify a road network, and the other determines the shape of the city boundary.

Image inputs:

Two image files are used to define the terrain and zoning that the City Design plug-in will automatically produce in 3D. One gray-scale image is used for specifying terrain data. The other image determines the layout of city zones. A user creates the size and shape of zones by painting color coded areas with an image editor like Adobe Photoshop or Fireworks. Each color is associated to one specific procedural modeling process such as rural, suburban, and urban district generation. Both the terrain and zoning image files must share the same pixel dimensions. Based on widespread availability in academic and professional settings, input file types were chosen that can be created using the Adobe products Illustrator and Photoshop.

City model generation in 3D Studio Max:

The City Design plug-in for 3D Studio Max interprets data from the four user generated input files to automatically produce a 3D model. The plug-in, implemented in MaxScript, begins the modeling process by creating terrain, street network, and parcel data extracted from the input files. Using procedural modeling techniques, building geometry for each parcel is then generated. The color-coded zoning input file previously created by the user determines what kind of building is generated on each parcel (rural, suburban, or urban core). A database of prebuilt foliage models (see fig. 3) is then used to landscape each parcel and street.

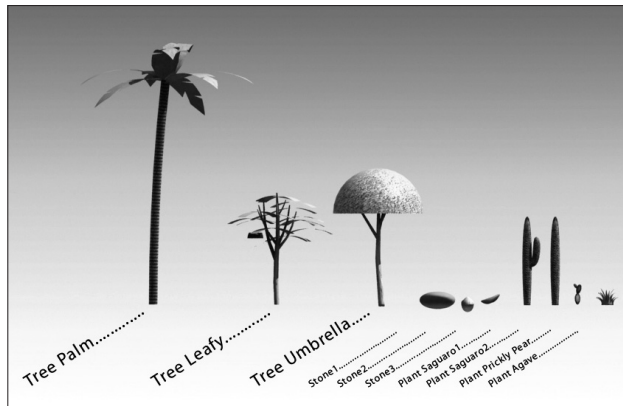


Figure 3. Predefined 3D Components.

After modeling, each parcel is automatically packaged as a single object including ground, sidewalk, landscaping, and building(s). A multi-sub object texture from a predefined database is then selected at random and applied to the object. The textures share a similar style, but add variation to building models in color, building and roofing materials, and ground texture.

Once all modeling and texturing is completed for each parcel in the scene, the textures are rendered and baked. Our test model uses the V-Ray rendering engine, but the City Design plug-in accommodates the use of Mental Ray as well (Chaos Group, 2009). Figure 4 shows the output of several parcel variations for an urban core zone using V-Ray. All models are then exported from 3D Studio Max as .3ds files and are packaged with their corresponding texture file.

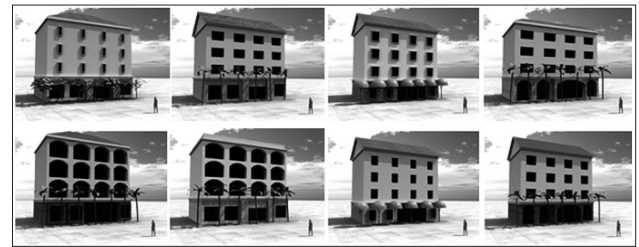


Figure 4. Generated Buildings by Procedural Modeling.

Visualization in virtual reality environment:

A separate tool in this study was developed to import the automatically generated urban environments into a VR package called UC-win/Road (Forum8, 2009). The real-time 3D environment allows street networks, traffic simulations, pedestrian movements, and other visualization features to enhance the city model. The plug-in tool enables the user to easily import the packaged model from City Design into UC-win/Road and was developed using Delphi 2007 (Cantú, 2007).

Results

This research has yielded several computational tools that allow urban designers, academics, students, and general users of 3D modeling packages to automatically generate fully textured and rendered 3D cities. An initial plug-in for 3D Studio Max has been



Figure 5. Screen capture of real-time test model.

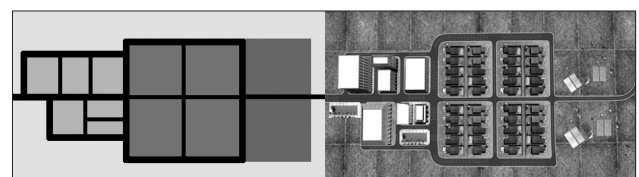


Figure 6. Input zoning file (left) and final output (right).

SIGraDi 2009 sp

developed that allows the user to create a city model using only 4 input files - two drawing files and two images files. The plug-in exports and packages the 3D city model and associated baked textures as .3ds files and .jpg images. A supplemental application was created using Delphi 2007 that allows the completed city model to be demonstrated in a real-time VR environment.

Utilizing City Design plug-in with test model:

The development efforts presented in this study yielded an automatically generated, textured, and exported test model. The test model was created by a single user in a fraction of the time it would normally take a team of modelers to complete using conventional methods. Using the supplemental tool developed in this study, the model was then imported into UC-win/Road, a VR software package in which a fully editable and drivable street network was added (see Fig. 5).

The proof-of-concept version of City Design used to automatically generate the test model in this study contains the functionality to develop environments with three zones: rural, suburban, and urban. Future versions will be expanded to include all six primary zones included in the SmartCode transect. Figure 6 illustrates the initial input used in the City Design plug-in and the final output. The left image is a combination of the zoning and street network input files and the right image is a screen capture of the finished VR model generated using City Design.

The presentation of the test model in UC-win/Road demonstrates a successful completion of this procedural modeling framework. The test user was able to create four input files using Adobe Photoshop and Illustrator, use the inputs to automatically generate a city model, then import the packaged model into a virtual reality environment.

Conclusion and Future Work

The design tools presented in this paper are the initial steps in a plan for significantly broader development. As previously stated, future versions will allow users to automatically generate cities with all six zones based on the SmartCode transect. Additionally, different styles of building models, foliage, and textures will be implemented to give users more options in creating cities for different regions or complexities.

This research has laid the groundwork for developing a versatile tool that can have valuable impacts in several industries. The automated modeling, texturing, and rendering process provided by City Design fits within the suite of software applications already used by many 3D modeling, design, architecture, gaming, and movie studios, allowing these tools to be implemented with little cost and training. Additionally, the ability to rapidly generate city models based on SmartCode will provide urban designers and city planners with a powerful and future-minded visualization tool.

References

- Autodesk (2009). 3DS Max 3D Animation and Rendering Software. Retrieved September 04, 2009, from <http://usa.autodesk.com/adsk/servlet/pc/index?id=13567410&siteID=123112>
- Bruneton, E. (2005). Modeling and Rendering Rama. Retrieved September 09, 2009, from <http://ebruneton.free.fr/rama3/rama.pdf>
- Cantù, M. (2007). Delphi 2007 Handbook. Italy: Marco Cantù.
- Chaos Group (2009): Major Features of V-Ray for 3dsmax. Retrieved September 02, 2009, from http://www.chaosgroup.com/en/2/vray_features.html
- Duany, A. (2008). SmartCode Version 9 and Manual. Ithaca: New Urban New Publications.
- Forum8 (2009). UC-win/Road product information. Retrieved August 26, 2009, from <http://www.forum8.com/english/uc-win/ucwin-road-e1.htm#30403>
- Procedural, Inc. (2009). CityEngine Features. Retrieved Sept 04, 2009, from <http://www.procedural.com/cityengine/features.html>
- SmartCode (2009). Why we do this. Retrieved September 04, 2009, from <http://www.smartcodecentral.com/about.html>