Collaborative City Modeling

Erik Palmquist ¹, Jonathan Shaw ²
Georgia Institute of Technology, USA
http://coa.gatech.edu/imagine
¹ palmquist@coa.gatech.edu, ² jonathan.shaw@coa.gatech.edu

Abstract. This paper presents an approach to creating an online real time rendering environment, upon which a large-scale, urban 3D model can be produced as a collaborative effort between initial content creators and outside parties with an interest in simulation and visualization. In 2007, the City of Atlanta, Georgia organized a taskforce to provide recommendations on the future development and mobility along the city’s signature street, Peachtree Street. To aid in the visualization of this area, datasets were converted into low polygon textured 3D models for the entire study area. This content will serve as the foundation of a collaborative effort to complete a high quality real time environment. The process for this project will be described and the means to extend the boundaries, maintain, and collaborate with this content will be proposed.

Keywords: 3D model; collaborative design; real time; visualization; training.

Introduction

Recent articles in mainstream news and print have exposed the public to current research in real-time rendering environments. International coverage of the Architecture, Engineering and Construction, AEC, community’s use of Google Earth, Second Life, and commercial video game engines has introduced the public to the potential of this medium. In one such article, Vella (2007) reports on a leading architectural firm licensing Unreal Engine 3 for project visualization. Another column by Tardif (2008) reports on the topic of Google Earth, relating, “there was no reason why architects couldn’t model and submit their own buildings, and many reasons why they should”. The growth of the commercial video game market has broadened the age range of users that will have basic training in navigating and interacting with real-time rendering environments.

A Potential Collaboration

Commercial software has been developed to aid nearly every discipline within the AEC community. The academic and AEC communities use these powerful CAD and visualization software packages to create a complex set of content that support design and communication of projects. Each resulting software
package typically has its own binary file formats, scripting languages and software development kits for further development. Due to the need to share information between AEC communities, user communities have emerged around common conversion pipelines to promote collaboration between the disciplines. Most of the conversion pipelines preserve two and three dimensional geometric data and UVW coordinates the corresponding projections of two dimensional textures on the geometric data, between software packages at a minimum. Individually, each of these software packages can manage complex amounts of additional information and perform simulations on the content of a file, but this is typically lost in an exported file format accepted by another software package found within the AEC community. While some file formats and standards, such as Autodesk’s FBX/DAE and IFC, are being developed to preserve more of this data, we have yet to see them adopted by the software packages used by the AEC community.

Computer software products such as Google Earth/Sketch Up and Microsoft Virtual Earth/3dvia allow the public to contribute to a larger consolidation of textured 3d models and geographically tagged information. As Zeile et al. (2007) explain, “immersive technologies become more important, not only among experts in spatial planning, but also many private users are interested in these new tools.” User communities of these software packages have collaborated in authoring content, plug-in development, and reference data sets. The growing populations of these user communities have demonstrated sustained activity of collaboration through public contributions over the past few years. This is of particular importance, because as Day and Radford (1995) state, “building a computer city model makes necessary an ongoing commitment to maintain and update the model as the software and hardware which constitute the modeling environment changes and the city itself also develops”. Many municipal organizations have explored the logistics of transitioning from traditional 2d digital city maps for planning purposes into 3d (Arponen, 2002). Some of these organizations, like the Boston Redevelopment Authority, make datasets available to the public and consolidate contributions to create accurate 3d models of their cities. These products have proven that the public is willing to create and maintain a 3d representation of existing architecture and will tag these assets with additional information. Since these products are available for free, they lack the robust development of commercial packages used in the academic and AEC communities.

While these software tools may only be used in the AEC community to sketch or aid in presentations, the public’s interest and willingness to be trained in fairly simple software that can produce textured 3d models suggests a potential collaboration between these two groups. With the growing use of 3d visualization engines, content creation tools, and reference data sets made freely available to the public, both communities can collaborate toward creating a large scale real time rendering environment that can become a communications medium for the two groups.

The Video Game Industry

Since the late 1990’s some video game companies have found that by releasing an editor with a commercial video game the public will contribute content and extend the lifespan of a video game title. The PC first person shooters Quake and Unreal Tournament were two titles which witnessed the emergence of user communities that could contribute high quality 3d models and textures as well as tools for modifying the initial commercial content. The editor also exposed the commercial titles content, allowing the public to see how the content is put together for an interactive environment. Both titles released documentation of the pipelines for content created in other applications and optimization techniques for efficient environments. The AEC community has been evaluating the potential in using real-time visualization in conjunction with these video
There is a difference between the quality of content created by the public and the AEC community and the overlap in use of this content is minimal. The content created by the public may lack accuracy and consistency, which is necessary to be used more readily within the AEC community. The lack of accuracy and consistency can be addressed through accurate reference material and the establishment of simple standards in the creation of the content. The textured 3d models found in Google Earth and Microsoft’s Virtual Earth are typically low polygon shells of the building’s exterior with photography sourced textures that contain inconsistent lighting, angular distortion, and artifacts such as landscaping projected onto the building surface. Most dimensions of these 3d models are based on a distorted aerial photographs where the height is the most difficult to determine and typically the least accurate. The content created by the AEC community is overly detailed and complex for the most common uses of the models by the public. In order to bridge this gap, each group’s content needs to be brought up or down to some middle ground that can benefit both communities.

The video game industry can provide a platform for this middle ground that is needed to foster the collaboration between the public and the AEC community. With the research and development of graphics, physics, and artificial intelligence for video games, the AEC community can develop the conversion pipelines that preserve more data from their commercial software packages. The AEC community can use these powerful video game engines to communicate the results of complex simulations and visualize large data sets in a networked real-time rendering environment. As the video game industry grows and continues to research and develop tools that can be adapted for use in the AEC community, we can develop the real world environments for use in education, communication, and collaboration.

In the fall of 2006, we collaborated with the Peachtree Corridor Taskforce to aid in the visualization of their study and recommendations for the future development along the City of Atlanta’s signature street. We were asked to create a real-time environment of the entire study area and animations for the recommendations. We arranged the acquisition of LIDAR, aerial photography, and documentation of current developments for the entire study area for a reference dataset. We selected the UnrealEngine2 Runtime, a free educational variant of a commercial game engine, as the platform for the real-time environment due to the large study area and development of a custom interface and navigation system. Autodesk 3ds Max and Adobe Illustrator and Photoshop were used in all content creation.

The City of Atlanta’s Neighborhood Planning Units were used to derive unique naming prefixes numbered by block for the division of the reference datasets. With two semesters sponsored instruction for 14 students at 13 hours per week, we assigned each block for modeling and texturing. We established the requirements for naming conventions, level of detail, optimization, and color saturation of all the content created by the students. The students created 1250 models and textures along the 23.3 kilometers study area, approximately 9.3 square kilometers.

During the initial training for the students, we realized that some were more proficient with either modeling or texturing. We established two positions of responsibility for the students, modeler or texture artist. The students’ preferences were evenly split so we had the modelers pass their work on to the texture artists. This allowed us to complete a block with two assignments for the students. The training for the students focused on the modeling and texturing of low polygon 3d models optimized for real-time visualization.

In addition to the training of the students, the faculty involvement in this project was directed at
more complex tasks of automation routines for import into the Unreal Engine, complex modeling and texturing tasks, and customization of the interface and navigation. The automation routines were developed in the previously mention software application scripting languages, primarily Maxscript, Javascript, and Unrealscript. The complex modeling and texturing tasks were the roads, terrains, and highway overpasses. The customization of the interface and navigation was developed in Unrealscript utilizing some built in functionality of the game engine.

**Model Process**

The process for creating the 3d models was determined through the intended use by the Peachtree Corridor Taskforce of the real-time environment. They requested a helicopter height tour of the existing conditions along Peachtree Street for the entire study area. This request gave a range of detail to be modeled versus textured. A 1 meter level of detail requirement for the surface of the buildings was established as the completion point for the 3d models. Any surface details that were less than 1 meter would be represented in the texture. Another decision was to simplify the ground floor of the buildings and how they met the landscape, since the terrain and landscaping details were to be texture rather than geometry.

Each student was assigned blocks to model and was provided with the cropped LIDAR mesh, aerial photography, and oblique photography from at least two directions. They were encouraged to gather more photographic resources if needed. Starting from the LIDAR reference, a student could model the massing and add the 1 meter level of detail to

---

**Figure 1**

*Model Process*
the surface from analyzing photography and LIDAR together. The resolution of the LIDAR could give a few points for reference at 1 meter but needed to be visually confirmed through the photography. Once students completed a model of each building, they went through a model completion checklist to ensure that the naming conventions, optimization routines, and ‘water tight’ requirements were followed (Figure 1).

Once all the buildings were modeled they would turn in their assigned block and start on the next one. The completed block would be reassigned to a student for texturing.

**Texture Process**

The students involved in the texture process received the same resource materials as well as any contributions made by the modeler. The first step for each 3d model was the unwrapping of the UVW coordinates into a two-dimensional square texture. Students were trained to balance the optimization of the texture coordinate space while coordinating adjacent elements of the surfaces. The unwrapping of the UVW coordinates also served as the final ‘water tight’ geometry test by identifying overlapped and inverted faces. Any unexplained overlap or inversion of a face would identify the problem area in the mesh and could be removed or corrected before the UVW coordinates were finalized.

After rendering out a UVW template from 3d Studio Max, the students would import this template into Adobe Illustrator. The students were trained to create vector elements to represent the façade details below 1 meter and the colors for the material finishes. Commonly used elements, such as windows, doors, and rooftop mechanical systems would be saved into symbol libraries for sharing among all the texture artists. Simple depth cues for the 1 meter details could be achieved through inner or outer shadows on the vector elements. After a final checklist of color saturation, optimization, and level of detail, final Adobe Illustrator files were saved for later processing.

After the completion of all the 3d models and UVW unwraps, we could render lighting simulations using the same UVW coordinates. We used 3d Studio Max day lighting system and the mental ray renderer for the lighting simulation. The raster texture size was automatically determined by each building’s surface area, a residential home would be around 128x128 pixels and a skyscraper would have a 1024x1024 texture. This allowed a consistency in visual detailing based on the surface area and was composited in Adobe Photoshop with the diffuse Adobe Illustrator file (Figure 2).
The final texture for the building was imported and linked to the 3d model in the UnrealRuntime2 Engine.

**The Automation Process**

Since we had to process over a 1200 models and textures for import into the UnrealRuntime2 Engine, we developed routines to automate most of the tasks. A combination of scripting languages were used to write simple scripts to automatically process, place, and link the 3d models and textures. The development of this automation was done as needed along the conversion pipeline. Scripts to preserve location, batch export file formats, and link textures were developed in Maxscript for 3d Studio Max. Javascript was used to automate the composition of the diffuse and lighting textures in Adobe Photoshop. Unrealscript was used to link the textures to the models and automatically set placement and default properties in the UnrealRuntime2 Engine. At each location along the conversion pipeline we attempted to preserve all necessary information that would otherwise get lost in the existing file formats and imports.

The importance of establishing unique naming conventions for the organization of each block allowed this automation to be successful. We were able to leverage existing standards developed by the City of Atlanta and adapt it to organize and distribute this project with a relatively small group of students and faculty in less than a year. The automation routines developed specifically for this conversion pipeline can be easily adapted for other applications, such as Google Earth or VISSIM. As AEC commercial software packages and file formats continue to preserve more information, the development of conversion pipelines into real-time rendering engines will need further development or collaboration.
Results

After the completion of all the models and textures for the buildings, roads, and terrain, the content was imported into the UnrealRuntime2 Engine. An aerial view of the downtown area can be seen in figure 3. A simple navigation method was created to allow any user to easily navigate the real-time rendering environment. We decided to exclusively use the mouse, since most people are not familiar with a traditional PC video game navigation utilizing the keyboard and mouse. The user can control the view through the mouse movement and move along Peachtree Street by holding the left mouse button. An alternative, location to location mode could be entered with the right mouse button. This mode would highlight the current location and move to a higher altitude and allow an arrow preview of the destination (figure 4). Once the destination is determined the left mouse button confirms the selection and the camera will return to the current destination and automatically move the user to the destination allowing the user to control the view while moving. A simple heads up display, or HUD, was developed to provide direction, speed of travel, ground elevation, and a map of the study area. An additional layer of information was added for the taskforce to show areas that are susceptible to change in the next 5 and 10 years. These areas were represented with tinted, red for 5 years and blue for 10 years, transparent volumes that were extruded to the future land use heights.

Since completion, the resources and content created through this project have been used in College of Architecture studio courses, traffic simulations, and internally with Georgia Tech. We have connected our Georgia Tech campus model to the Peachtree Corridor model. We continue to develop the UnrealRuntime2 Engine build to include networked collaborative design tools, pedestrian way finding and traffic simulations, environmental access, and
facilities asset tracking for the Georgia Tech Campus. We have found our content is easily imported into Google Earth, mobile phones, and some AEC simulation software. We have also established traffic simulation results from VISSIM into our UnrealRuntime2 Engine build. The developments since the completion of this project have shown that the content can be shared and distributed in a variety of formats and can foster collaboration between the public and the AEC community.

**Conclusion**

This release of this project can initiate collaboration between the public and the AEC community to create an accurate and consistent real-time visualization environment. The two groups with differing experience levels and various software tools can streamline their efforts towards the creation and maintenance of 3d content. With the proper datasets, training, and efficient pipelines, large-scale urban models such as this can be created. This environment will continue to expand over time into an even larger metro Atlanta model and begin integrating simulation results, such as lighting, traffic, and shadow studies.

Its potential to be utilized by the larger community will be a focus for further study. The process that was developed during for the Peachtree Corridor is applicable to other city modeling projects which call for an element of cooperation between different groups. Many municipal organizations are collecting the needed resource data sets, such as LIDAR and aerial photography, to start such an initial effort for other cities. Collaboration between local academic institutions, AEC members, and the municipal organizations can begin creating a base city model that can be used by the entire metropolitan community.

**Acknowledgements**

The authors wish to thank the late Dr. Thomas D. Galloway, the members of the Peachtree Corridor Taskforce, and the members of the IMAGINE Lab including, Dr. Anatoliusz Lesniewski, Matthew Swarts, and Racel Williams, and all of the graduate research assistants who were involved in this project. We would also like to thank the City of Atlanta, Georgia Power, Urban Collage, and Kimley-Horn and Associates, Inc. for collaborating and assistance in acquiring our resource data sets.

**References**


