A USER INTERFACE FOR PARAMETRIC ARCHITECTURAL DESIGN REVIEWS

JAWAD ALTABTABAI¹ and WEI YAN²
¹,² Texas A&M University, College Station, USA
{jtabtabai, weyan}@tamu.edu

Abstract. Architectural form and performance are affected by the designer's graphical representation methods. Parametric CAD systems, as design and representation tools, have become ubiquitous in architectural practice and education. Literature in the area of parametric design reviews is scarce and focused within building inspection and construction coordination domains. Additionally, platforms marketed as design review tools lack basic functionality for conducting comprehensive, parametric, and performance-based reviews. We have developed a user interface prototype where geometric and non-geometric information of a Building Information Model were translated into an interactive gaming environment. The interface allows simultaneous occupation and simulation of spatial geometry, enabling the user to engage with object parameters, as well as, performance-based, perspectival, diagrammatic, and orthographic representations for total spatial and performance comprehension.

Keywords. Design cognition; Virtual/augmented reality and interactive environments; Human-computer interaction.

1. Introduction

The way in which we visualize architecture, directly by graphical representation methods, affects the way we design and perceive architectural form and performance (As and Schodek, 2008). Generative and parametric design methodologies have become ubiquitous in architectural pedagogy and practice (Aish and Woodbury, 2005). The new generations of parametric CAD systems allow for quick production of variations in design (Erhan et al, 2009). The complexity of generative architectural design stems from the large interrelated parameters. The problem with parametric/generative modelling is the inability to visualize and present a project containing many de-
sign options by changing the object and relational parameters in real-time. This lack of real-time form generation conceals the parametric framework of a given project, thus forcing the designer to present a static model of a parametric and dynamically conceived design.

Currently, there are multiple platforms, which are designed as presentation tools for parametric models. Usually, these platforms have a simplified interface, which lacks parametric information and completely disregards performance-based visualization.

Given the malleability of in-game interface design, game-engines may provide an opportunity for the creation of an interactive, parametric, and to some extent, performance oriented experience of architectural projects which may alleviate the problems of presenting parametric models for design review. As an interactive tool, game engines could enhance spatial awareness, which may result in informed design decisions (Hoon and Kehoe, 2003).

This research introduces the study and creation of a prototype for a dynamic presentation method for parametric and performance-based models that can evolve in real-time. The prototype can be achieved by translating architectural design into a gaming environment and through the design of a novel user interface that simultaneously engages object parameters, along with performance-based, perspectival, diagrammatic, and orthographic representations.

2. Literature Review

Research in the area of design reviews for parametric and Building Information Models (BIM) is scarce. Collectively, parametric modelling software lack functionalities for design evaluation. For design review purposes, Kumar et al (2010) developed an interface for experience-based design review for health facilities. Shiratuddin and Thabet (2011) explored ideas of a design review interface for non-parametric models. More related research is found within the domain of architecture, engineering and construction (AEC), specifically for building design inspection and coordination. Keough et al (2009) developed a mobile application for collaboration through BIM model viewing, mark-up, and data-query. Pauwels et al (2010) created a test case to illustrate how gaming engines and architectural semantic web can be combined to enhance information visualization for architectural design and construction. Yan et al (2011) developed the BIM-Game prototype that integrates BIM and the XNA® gaming engine for enhanced design visualization. The limitations of the mentioned research examples stem from the confinement of screen space to desktop and mobile platforms and from the lack of performance based visualization.
Currently, there are platforms marketed as design review tools for parametric models. For example, Autodesk Showcase® is a design review tool that allows for animation and interactive walk-throughs. BIMx® is a desktop and mobile platform for the exploration of BIM models with game-like navigation for presentations and construction coordination. Lumion3D® is a real-time rendering engine that can export still images and choreographed walkthroughs. Similarly, LumenRT® is another platform that allows for a live walkthrough mode with the ability of displaying limited BIM information if available.

The limitation of these platforms stems from their simplified interface, which prevents customization, lacks parametric properties, and completely disregards performance-based visualization. Almost collectively, these platforms restrict the interaction with the model to a singular viewport, preventing simultaneous experience of orthographic drawings and perspective views. Therefore, the designer is constrained by the preconfigured software package that requires switching between different views to gain total comprehension of spatial qualities, and switching between different platforms to view the parametric information and performance-based visualizations.

As architecture deviates from traditional forms, forms that are harder to express using projective drawings, representation shifts from dictating architecture to provide an interface for decision making (As and Schodek, 2008). We propose overcoming the limitations of the above platforms by prototyping an interface for immersive, parametric, and performance oriented real-time experience of architectural projects. The prototype is a tool for studying and evaluating the effectiveness and performance of a design project. Because daylighting is often considered as a major component of green building design (Leslie et al, 2012), we made the decision to include daylighting analysis as a performance-based example for the proposed interface.

3. Building Information Modelling and Game Engines

BIM is a comprehensive database for geometric and non-geometric information, which facilitates the creation and management of AEC and operation data throughout the lifecycle of a building from design to post-occupancy. BIM’s parametric constraints and relationships between model elements allow for quick iterations of design updates. Due to our cumulative knowledge of Revit®, we have made the decision to use it as the BIM platform and use Dynamo® as the visual scripting environment throughout this research.

A game engine is a platform for the creation of interactive environments. Gaming engines contain an integrated collection of modules that are necessary to simulate the interactive environment. The possibilities for interaction
in a virtual gaming world are limitless where as a medium, gaming engines can simulate nearly every other type of media (Gauthier, 2005).

In virtual gaming environments, buildings are often modeled and textured. This is where using a BIM platform becomes an advantage, as a high level of comprehensiveness and information required for an interactive environment can be rationalized and embedded in BIM models (Hoon and Kehoe, 2003). Because of our familiarity with, and the abundance availability of references for Unity3D®, we have made the decision to create the prototype using the latest professional beta version of Unity 5.0.

4. From BIM to Gaming

4.1. BASIC ASSUMPTIONS

Due to the fact that exported FBX® files from Revit have encrypted textural properties, we have made the basic assumption that only a single texture map may be assigned to each element. For example, there are no representational differences between the interior and exterior finishes of a wall. Although it is possible to retain textural properties from Revit to Unity via Autodesk 3DSMax®, and via the use of commercial plug-ins (Bille et al, 2014), a decision was made to overlook this limitation and explore it in future research.

4.2. EXPORTING 3D GEOMETRY, MASSING AND STILL IMAGES

To create an interactive environment of the BIM model in Unity, we created a series of simple Dynamo scripts to assist in the automation and efficiency of the export process from Revit. We exported the polygonal geometry in FBX format from Revit into Unity via a custom Python script in Dynamo. The script exports multiple and separate 3D views simultaneously, which allows us to save different parts of the model, such as architectural elements and MEP separately, allowing better object management in Unity.

Another script was created to export the bounds of each room. Because Revit does not create geometric representations of room bounds, the script first generates the geometric bounds and imports them into the Revit model before exporting them as a separate FBX file. The function of these bounds will become evident when we explain the interface in detail in Section 5.2.6. Different 3D views and work-plane daylighting distribution analyses were rendered using Autodesk360® Cloud rendering service called from Dynamo.

4.3. EXPORTING OBJECT PARAMETERS

For a design review, not all parametric elements or properties are useful or needed. Therefore, we created a custom script, which only considers the par-
ametric properties of elements specified by the user. The script cycles through each selected element and exports its properties from Revit into a separate text file (parameter file) that will be imported into Unity for presentation of object parameters.

4.4. IMPORTING REVIT INFORMATION INTO UNITY

Once geometric definitions, parameters, and images are stored into the Unity project, we can situate the model, texture it, define collision properties, and attach different Unity scripts to allow for interactivity. For example, we retrieve the parameters of a window by attaching a script to the object we want to query. The script searches for the object’s parameters in the parameter file using the Revit objectID and displays the information in the Unity viewport. This functionality is possible because each element in the FBX file retains the Revit objectID as the element’s name suffix (Pauwels et al, 2010).

5. Interface implementation

5.1. PROTOTYPE GENERAL DESCRIPTION AND ASSUMPTIONS

The basic assumption for the interface is that the screen space is unlimited. Therefore we are not confined by the size of a particular monitor or projector. The interface is divided into two main categories, a main viewport and a series of support viewports (Figure 1).

The main viewport is a first-person perspective camera for real-time navigation of the project (Figure 2). The main viewport is rendered using Unity’s newly integrated Enlighten engine, which is a dynamic real-time global illumination (GI) solution (Unity Blog, 2014). Support viewports are completely customizable where the designer chooses relevant information and visualizations for the design review (Figure 1). To assist the user in obtaining accurate spatial perception in the navigable environment, the location of the user should always be highlighted in the support viewports (Fukatsu et al, 1998). Furthermore, the path a user takes during the design review can also be represented by drawing a line of his/her journey in space (Yan, 2006; Yan et al, 2007).

Figure 1: main viewport 1 and customizable support viewports 2-9.
5.2. PROTOTYPE DETAILED DESCRIPTION

5.2.1 main viewport

In addition to the general assumptions, as an example, the user can toggle a work plane illuminance distribution map in the main viewport for 9:00am or 3:00pm on September 21st. It is also possible to toggle a 3D false-colour (FC) overlay of the viewport. The script for the FC overlay samples every pixel in the viewport and maps the pixel’s brightness to a predefined colour gradient for providing an approximate 3D real-time illuminance visualization of the space. Additionally, the user can display parametric properties of certain elements in the model via a pop-up window. For this prototype, walls and windows parameters were incorporated for proof of concept.

Figure 2. Examples of the main viewport functionality. From left to right: GI and parameters, GI and 9am illuminance distribution map, GI and 3pm illuminance distribution map, 3D False Colour approximation map for illuminance.

5.2.2 Support viewport: mini-map

A mini-map is a term borrowed from the gaming industry, which is a small viewport usually placed in the corner of the screen. The mini-map is a dynamic top view of the user in space for orientation assistance. The mini-map inherits its transformation from the user’s movement; therefore, if the user moves horizontally in the space or vertically between different levels, the mini-map will update to reflect the user’s position (Figure 3).

Figure 3. Different examples of the mini-map viewport. Note: the map inherits the users rotation values in space where the camera frustum always points upward on the map.
5.2.3 Support viewport plan and section

Plans and sections are dynamic real-time viewports generated using Unity’s camera clipping planes and a specific clipping shader for the wall poche. As the user moves to a different level, the plan viewport will automatically update to display the appropriate drawing at the user’s location. Similarly, the section viewport constantly provides feedback about the spaces occupied by the user and the adjacency of the spaces (Figure 4).

![Figure 4. Examples of Unity generated dynamic plans (top) and sections (bottom).](image)

5.2.4 Support viewport still renderings

This viewport provides an opportunity to evaluate the model using Autodesk360 cloud rendering service where the user can view predefined realistic renderings of spaces on September 21st 9am and 3pm (as an example). Additionally, one can cycle between realistic renderings and FC renderings to further evaluate the lighting distribution in a particular space (Figure 5).

![Figure 5. Still renderings generated in Revit via Autodesk360 and imported into Unity to be displayed in the support viewports.](image)

5.2.5 Support viewport illuminance distribution

We have provided an independent support viewport for inspecting the work-plane illuminance (Figure 6) for an overall comprehension of daylighting
distribution. Similar to the plan and section viewports, the illuminance distribution viewport dynamically displays appropriate information considering the user's location in space at user specified times.

![Figure 6. The illuminance distribution viewport on September 21st at 9am and 3pm](image)

5.2.6 Support viewport axonometric massing

The axonometric massing can detect when the user enters a space then highlights the space with a different colour (Figure 7). The addition of this viewport assists the user in gaining insight into the general massing of the project. Because occlusion may occur where the user is hidden due to overlapping of masses, the viewport has the ability to rotate in an increment of 45 degrees.

![Figure 7. Different possible configurations of the axonometric massing viewport.](image)

5.2.7 Complete Interface

The current complete user interface is displayed in Figure 8.

![Figure 8. The completed interface prototype showing the main viewport (with the axonometric massing viewport and parameter viewer) on the left and support viewports on the right.](image)
6. Discussions

Note that this case study is made only for one of the design options of a project. However, it enables the following parametric design review process: When design changes, new design options will be generated from parametric BIM models, and corresponding daylighting performance results will be updated. Both the new design option and the updated performance results will be presented in the prototyped viewports. The complete design update – presentation update process will be in real-time, which will be supported by pre-simulation of the daylighting performance for all predetermined design options. This will allow for different design options and associated performance analyses to be reviewed during the design review, which will lead to informed design decisions.

To facilitate the design option selection process, an optimization process can be integrated into the pre-simulation of building performance using the parametric BIM models. For example, we can conduct multi-objective optimization for design objectives that may include building functions, energy and daylighting performance, costs, etc. and compute the set of optimal solutions (design options) – Pareto Optimal Set (Radford and Gero, 1987). There can be multiple design options, each of which meets the following condition: any improvement of one objective will degrade at least one other objective. These design options are the Pareto Optimal Set.

In the design review process, the Pareto Optimal solutions can be presented parametrically for both the building forms and performance. The review will focus on the creativity of designs’ performance trade-offs, while all the presented design options are optimal in terms of performance.

7. Conclusion and future work

In this paper we presented the first phase of the development of a user interface for parametric architectural design reviews. The first phase includes the transference of geometric and non-geometric data from Revit into Unity, and a working prototype where the user can interact with an integrated environment of geometric, parametric, and performance-based representations. The prototype is valuable in gaining comprehensive understanding of the project by juxtaposing multiple viewports, which dynamically update based on the user’s spatial location.

Future work will include a case study for a complete parametric design presentation that is described in Section 6 Discussions. It will also include a thorough user study for the evaluation of the interface. Using validated evaluation methods, we will conduct experiments for assessing the usability of the interface. If the effectiveness of the tool is validated, the proposed inter-
face may transform and enhance the way in which we present parametric models for design review. It will provide a fertile ground for simultaneous occupation and simulation of spatial geometry. Additionally, the interface in its current state presents parametric properties of singular elements. However, parametric design can be very complex where elements may have their own parameters in addition to relational parameters and dependencies with other elements. We are currently exploring the possibility of exposing the complex relationships of parametric models in the prototyped interface.

References


Shiratuddin, M. F., and Thabet, W.: 2011, Utilizing a 3D game engine to develop a virtual design review system, ITcon, Vol. 16, 39–68.


