Abstract. Digitally-generated visualizations, such as renders or movies, are, nowadays, commonly used as representation methods for architectural creations. This occurs not only in final stages of the process, with the goal of selling the product’s image, but also in midst creation process to express concepts and ideas. Presently, the spread of parametric and algorithmic approaches to design creates a problem for visualization, as it enables the almost effortless change of 3D models, thus requiring repeated visualization efforts to keep up with the changes applied to the design. To solve this, we propose extending the algorithmic design approach to also include the high-level description of architectural image creation. The methodology, Algorithmic Architectural Visualization (AAV), also contemplates the required preparation settings for the visualization process, and includes possible visualization productions inspired by film techniques.

Keywords. Algorithmic Design; Architectural Visualization; Render; Film Grammar.

1. Architectural Visualizations

Selling the project is a fundamental task in architectural practice nowadays and allowing the client to experience that project, beyond the typical 2D representations, is an important step to achieving it. Digitally generated images offer “visual experiences” (Correia et al., 2013) of yet unfinished projects, giving the viewer a better perception of how the spaces will be. Hence, clients are now ever more eager for this kind of representation methods.

These visualizations, commonly denominated as renders, may consist of isolated pictures, or sequences of images. When intending to mimic the experience one would have in a built space, a dynamic perception serves the purpose far better than an imitation of a static gaze (Correia et al., 2013). Thus, an architectural visualization package should include, not only static image sets, but also animations of the projects, rendered films that can give life to the unbuilt architectural creation (Ng, Schnabel and Kvan, 2006).

1.1. VISUALIZATION PRODUCTION

Producing this sort of images is not a fast endeavor, particularly if photographic realism is intended. Traditionally, the production of presentation-oriented 3D
models used to be employed at later development stages, when more certainty about the project was already achieved. This was mostly due to the cost and the inherent difficulties of such models. More recently, however, with the evolution of the tools involved, we find these processes trivialized.

Many digital tools are available to create scalable, navigable, and sometimes interactive rendered images. Current architectural rendering applications are equipped with lighting and texturing features capable of giving depth and realism to the scenarios created. Most 3D modeling tools, within the CAD and BIM paradigms, have been incorporating rendering engines over the past decade. SketchUP, Rhinoceros, ArchiCAD, Revit, Maya, AutoCAD, Vectorworks are some examples. These tools allow the user to easily produce the 3D model, enhance its scenario, and render it within the same environment. Some of them also support plug-ins for that purpose, like V-ray or Maxwell.

The most highly regarded 3D Rendering Software, however, are tools like Blender, Modo, Cinema 4D or 3DS Max, which are specialized for the task. These tools have the rendering engine working in cooperation with the 3D modeling software, which means the user can model and render in the same environment as well. They also enable imports from other programs, allowing the user to model their geometry in another software and still benefit from the rendering features.

1.2. ALGORITHMIC PROBLEM

Algorithmic Design (AD), a method nowadays increasingly present in the architectural practice, defines the creation of forms through algorithms (Gerber and Ibañez, 2015). Using AD, the architect does not build the model directly, but instead, builds the program that builds the digital model, through a combination of geometric, as well as symbolic and mathematical representations of objects (Woodbury, 2010). An algorithmically designed project benefits from a great flexibility: since the design entities are logically connected, changes applied to the parameters are automatically propagated to the whole model (Burry, 2013), allowing the user to explore a variety of ideas with no extra modeling effort.

The spread of this design approach creates a problem for visualization. By enabling the almost effortless change of 3D models, ideally, the architect is required to repeat the visualization efforts for each iteration. Since the process of creating a rendered image is still far from fast, the prospect is not yet quite achievable.

The computer graphics community, driven either by animation or gaming issues, has been making progress in automating the tasks required to render a scene. Prima, et al. (2016), Lino and Christie (2015), and Burelli, et al. (2008) present interesting solutions on camera motion control, automatic placement of cameras in virtual scenes and generation of smooth camera paths. Zheng and Zheng (2017) propose a camera lens model capable of emulating the imaging of a wide range of real camera lenses. Other advancements include work on specific techniques for object perception in scene, such as motion blur rendering (Navarro, Serón and Gutierrez, 2011) or depth of field effects (Schott, et al., 2011). Assuming the architect develops his project using AD, the following sections propose the
Algorithmic Architectural Visualization (AAV) is here presented as a methodology for the effortless generation of adaptable visualizations. With AAV we intend, primarily, to demonstrate that the benefits of AD also apply to the visualization task. AAV relies on the inclusion, alongside the model’s description, of parametric descriptions of the rendering tasks that follow the building’s parametric description. This means camera positions and alignments follow the logic of the project, so that changes to the design entail corresponding changes in the visualizations.

The methodology contemplates the required preparation settings for the visualization process, namely detailing the model and the respective scenario, and it proposes possible visualization productions that may prove of interest to sell an architectural project, including construction sequences, walkthroughs, and dynamic shots. The latter includes the use of film techniques, which can help convey specific emotions the architect may desire for the spaces in display.

2.1. MODEL PREPARATION

The first contemplated task by the AAV methodology, the preparations for an algorithmic rendering process, can be divided in two parts: (1) detailing the model to create ambiances, which may imply the definition of furniture elements, coatings, lights, etc.; and (2) defining scenario details, such as sunlight, sky, and other environment settings. Both tasks can only be programmed and automated as far as the rendering software in use allows, and the level of detail depends on both the project’s development stage and the purpose of the render itself.

An important aspect to consider, in either case, is the extension of the parametric flexibility of the base model to the detail added for rendering purposes. Since the spaces defined in the program are flexible and receptive to changes, the objects must be as well. Furniture and lighting, for instance, must be flexible in size and/or positioning, to change in accordance to the variations of the spaces. Ideally, generic algorithms for the placement of furniture could be used for each type of space, freeing the user from this tiresome task (Merrell et al., 2011).

2.2. VISUALIZATIONS

Next comes the decisions as to what image, or collection of them, best describes the idealized project. The typical selling pack includes a series of static shots from various angles to show the different aspects of the project, or dynamic sequences - films or animations, having the unbuilt architecture as the central character of a spatial narrative (Ng, Schnabel and Kvan, 2006).

In order to create a narrative that vividly represents the unbuilt forms and intrigues the audience, the author must consider not only the aspects of his creation to show, but also the manner in which he will show them (Ng, Schnabel and Kvan, 2006). The following subsections present some of the essential filming techniques architects can apply to the making of their own architectural visualizations.
2.2.1. Film Grammar

The Film Grammar provides us with both stylistic and technical rules, designed to convey specific ideas and emotions through shooting techniques and camera control graphics (Palamidese, 1996). In AAV, we propose the exploration of dynamic shots, such as (1) tracking, (2) panning, and (3) zooming. Each of these shots can be performed in a series of ways, which means a multitude of possible scenes can be created.

For instance, (1) tracking shots can be forward or backward, rightward or leftward, oblique, upward or downward, or even circular, and they are used to follow a particular subject or object as it moves (Arijon, 1991). (2) Panning means rotating a fixed camera around its support, and this can be done horizontally, vertically or obliquely. A pan can reveal additional information in the space or connect two points in this space (Palamidese, 1996). The conjugation of this technique with that of static shots, such as up-shots or down-shots, can also emphasize particular aspects of the project, by playing the feelings of dominance and inferiority. With (3) zooming shots, the viewer can either be pulled in or out of the scene, in the first case, by focusing on an object in scene, and in the second, by being revealed new parts of the scene.

AAV provides the user with the basic programming operations for these essential filming techniques. It is up to the architects to adapt such operations to their project’s algorithmic description in order to generate the film sequence that best displays the projects’ identity and features.

2.2.2. Walkthroughs

One common type of image sequence is the walkthrough. Composed of a sequence of shots taken at the human eye level and following a defined path along the project, it attempts to imitate the sensation the user could have walking through the project himself. The algorithm used for walkthroughs is somehow similar to that of the tracking technique, only following a pre-defined path instead of a specific object in the scene.

In order for any of the previously described sequences to be correctly produced for any instance of the model, users must guarantee the algorithms defining the paths are as flexible as the model itself. Only then can AAV’s operations adjust to the mutation of the spaces that results from changing the model’s parameters.

2.2.3. Animations

Depending on the context, it may also be relevant to produce conceptual animations. A (1) construction sequence, for instance, may be relevant to show the assembly chain that needs to be carried out during construction. The model generated for this purpose is usually focused on construction detail, excluding furnishing and other fine points of less interest to the matter. The production of the images must be intercalated with the generation of the model step by step, so that each new frame contains a new added element, until the whole assembly is complete.
The initially envisioned shape can suffer many changes along the project’s development, not only on account of the concept’s evolution, but also due to structural, economic, ecological or other reasons. Hence, the initial concept should be flexible enough to accommodate these demands without losing its essence. A (2) shape/concept flexibility animation serves to present the shape variations the algorithms allow, or, in other words, the possible variations to the shape the architect was willing to consider. This sequence of images requires a regeneration of the model with new parameters for each given frame.

These two types of animations are highly dependent on the project’s shape, concept, construction techniques, etc. The algorithms necessary to produce them must be constructed alongside the algorithmic description of each different project and, thus, they are illustrative examples of the AAV methodology.

3. Evaluation

For the evaluation of AAV we chose to model a case study project whose shape benefited from AD, the Astana National Library from BIG architects. Regarding the visualizations, we focused on production renders - a conceptual, non-photorealistic, stylized kind of visualization, commonly used in conceptual phases of the project, where algorithmic flexibility comes into play.

We modeled the case study in Rosetta (Leitão and Lopes, 2011; Feist et al., 2016), an AD tool that allows the design to be generated in a series of back-ends, both CAD and BIM tools. For most of the AAV evaluation, we used one of the BIM back-ends: ArchiCAD. This tool not only presents a user-friendly rendering engine - CineRender - that requires very few settings to obtain good results, but also possesses a set of pre-modeled objects useful to decorate the interior spaces.

The films and animations produced during this evaluation process are most relevant from a cinematographic point of view. For this reason, although in the following sections we present selected shots to illustrate some of the produced scenes, the complete series is also available online: http://web.ist.utl.pt/antonio.menezes.leitao/AAV.

3.1. MODEL PREPARATION

Since a render is always a speculative scenario, one can envision a mathematical distribution of furniture and people that generates these elements automatically, in accordance to the building’s spatiality.

In Astana’s case, we decided to take advantage of ArchiCAD’s pre-modeled objects and we programmed their distribution inside the library according to the buildings parameters. This means, if any changes to the model are made, e.g. in floor heights or wall distribution, the placement of these elements will change accordingly. Figure 1 presents two variations of the model (A and B), resultant from slight changes to its parameters. We also modified the available parameters of ArchiCAD’s pre-modeled objects to better suit the library’s environment. Figure 1 also presents two different furnishing options (C and D).
3.2. FILMS

Keeping with the principles of film grammar, and the techniques available, we conceived a storyboard for Astana’s rendering sequence that showed its essential and most fascinating aspects and features. Table 1 presents this plan, a storyboard composed of 9 scenes, and the following paragraphs explain its implementation.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Target</th>
<th>Camera</th>
<th>Location</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Dolly-zoom</td>
<td>Laptop placed on the table</td>
<td>Push forth: close in + widening lens</td>
<td>Middle ring library</td>
<td>Slowly reveal the space, not shifting focus from object</td>
</tr>
<tr>
<td>2 Tracking</td>
<td>Building’s center (static)</td>
<td>Bird’s-eye spiraling path</td>
<td>Outside, look inwards</td>
<td>Going around the building</td>
</tr>
<tr>
<td>3 Tracking</td>
<td>Linear path (moving)</td>
<td>Linear path (parallel mov.)</td>
<td>Outside, look inwards</td>
<td>Show the entrance arch to the inner patio</td>
</tr>
<tr>
<td>4 Walkthrough</td>
<td>Curvy path</td>
<td>Follow path</td>
<td>Inner circle &amp; middle ring</td>
<td>Walk at man height when possible. Fly over stairs.</td>
</tr>
<tr>
<td>5 Walkthrough</td>
<td>Straight path</td>
<td>Follow path</td>
<td>Outside, look inwards</td>
<td>Pass under the arch, enter inner patio, reach the middle</td>
</tr>
<tr>
<td>6 Panning</td>
<td>Straight path (moving up)</td>
<td>Building’s center (static)</td>
<td>Inner patio</td>
<td>Move from eye level to a low-angle shot</td>
</tr>
<tr>
<td>7 Tracking</td>
<td>Bird’s-eye curvy path</td>
<td>Follow path</td>
<td>Fly over top-floor, within the shell</td>
<td>Show interior space from above, inside the building</td>
</tr>
<tr>
<td>8 Panning</td>
<td>Top floor (moving left)</td>
<td>Building’s center (static)</td>
<td>Inner patio, look upwards</td>
<td>Show the view within the patio</td>
</tr>
<tr>
<td>9 Dolly-zoom</td>
<td>Laptop placed on table</td>
<td>Pull back: mov. away + closing lens</td>
<td>Middle ring library</td>
<td>Slowly remove background around the object of focus</td>
</tr>
</tbody>
</table>

AAV offers a predefined *tracking* function that receives a camera path and a fixed target, or a target path in alternative. In order to make scene 2, a sequence of bird’s-eye views of the building, we used the first version of the function. As
parameters, we provided the center of the building for the still target location and a spiraling path circling around the building with decreasing height for the camera. Figure 2A presents three shots belonging to this sequence.

The alternative version of the tracking function, a moving camera and a moving target, was used to produce the walkthrough sequences (scenes 4 and 5) and also scene 7. We algorithmically described a representative path running along the main public routes of the library for the target. The camera’s path is essentially identical to the target’s, only running a few positions behind. Figure 2B presents three selected shots from scene 4.

Another application of the tracking function was used to show the building skin from the outside (scene 3). In this case, the paths for the camera and target did not coincide, instead, they were parallel. We provided a linear path for the target and a vector, and a secondary function then created a similar path for the camera a short distance away from the original path, according to the given vector.

An also predefined function for panning was used, primarily, to transit from an eye level shot to a low-angle shot of the central atrium (scene 6), and secondly, to create a sequence of low-angle shots in the atrium as well (scene 8). Both sequences serve to emphasize the scale of the space. The panning function receives a fixed camera position, in this case, the center of the atrium at ground level, and a target path - a sequence of locations arranged vertically, in the first case, and in a circle around the atrium’s peak, in the second case.

The typical zooming techniques were also defined but not used in the Astana model. Inspired by iconic cinematic scenes such as the one in Hitchcock’s ‘Vertigo’, or Spielberg’s ‘Jaws’ and ‘E.T.’, we did, nevertheless, experiment with the dolly-zoom effect, which can be both revealing and disorienting, to give a dramatic beginning and end to our film (scenes 1 and 9).

Naturally, since the description of the model is parametric, we also programmed all the previously mentioned paths with enough flexibility to accompany the buildings possible mutations. Hence, if the building’s parameters
are changed, the defined paths will adjust accordingly, and new image sequences can be produced with no further alterations to the program. Figure 3 shows three possible outcomes for the buildings shape. Given a camera that runs along the building’s façade, following the Moebius strip’s twisting, the camera’s path automatically changes in accordance with the buildings shape.

Figure 3. Camera path (in red) for a ‘run along the façade’ film in three of Astana’s instances.

3.3. RENDER ACCELERATION

An additional advantage of AAV is the reduction of time and computational effort needed to produce the project’s films. For every new frame, a lot of effort is put into analyzing both the elements that appear in the shot, and those that do not. This occurs either when the rendering engine does not allow scene occlusion culling (Sudarsky and Gotsman, 1999), or when the existing culling mechanisms are inefficient in recognizing the fraction of visible geometry. In a model filled with decorating objects, this problem is particularly pressing.

The issue is solved in many of the rendering tools mentioned before. However, since ArchiCAD’s rendering engine is still at fault in this matter, we took advantage of the algorithmic approach to develop a culling solution of our own: the program divides all rendering sequences according to a predefined maximum number of renders per model and generates new models for each set. Each of these models contains only the indispensable elements for the frames in that set. Since the paths required for the AAV functions were defined with the same algorithmic logic as the building to film, it is fairly easy to calculate which areas of the project will or will not appear.

Naturally, the time required to regenerate the geometry is also considerable, thus, a trade-off must be found between the time gained in the analyzing process and the time lost in regenerating the model. The maximum number of frames parameter must, then, be adjusted according to the amount of detail in scene.

3.4. ANIMATIONS

To produce a construction sequence, we created an animated GIF containing the phased creation of all the main constructive elements of our case study (figure 4). For a shape/concept flexibility demonstration, we performed a video animation with possible variations of the Moebius strip. We varied parameters like the strip’s radius, height and length of its section, etc. The AAV methodology then allowed
us to compute the consecutive generation of the building’s variations, intercalated with the production of rendered images. The whole process is automated, meaning we had only to wait for the outcome - an animation sequence of the Moebius strip changing its shape.

Figure 4. Selected images from Astana’s construction sequence animation.

4. Conclusion
AAV not only automates most of the steps required to generate a visualization, but also ensures the flexibility of the endeavor. Since the entire process relies on the algorithmic description of both the model and the visualization settings, it is possible to effortlessly generate numerous variations of the model, along with the corresponding visualization sequences.

The proposed method also provides the user with the basic programming operations for various filming techniques, giving the architect the tools to easily create both static shots, animations, and films. This gives him the freedom to combine the production of the various possible scenes within the algorithmic description of his model, in order to generate the film sequences that best display the projects’ identity and features, thus largely improving the architect’s ability to express and promote his design.

4.1. FUTURE WORK
The render production presented in the evaluation of the methodology serves its purpose well in automating the architects’ task at non-final design stages, where the ambiance required is a generic one. However, for more advanced stages of scenario detailing, the approach cannot be applied on a basic rendering engine, such as ArchiCAD’s. For this reason, we plan on applying the methodology to specialized rendering software, by integrating new visualization backends to Rosetta, the AD tool used to test AAV.

Another interesting course of action involves the use of game engines, proven to offer a number of advantages when applied to architectural building scenarios (Pelosi, 2010; Koehler, Dieckmann and Russell, 2008; Hoon and Kehoe, 2003), including considerably reducing the overall time required to render multiple frames.

Acknowledgments
This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.
References


M. Burry (ed.): 2013, Scripting Cultures, John Wiley & Sons, Inc.


