Optimization Based Architecture Generation

Stewart Hills

Advisor: Norman I. Badler and Aline Normoyle
University of Pennsylvania

ABSTRACT

The recent popularity of procedural city generation in industry, has created a need for more specific tools. While tools exist to create buildings in single architectural styles or along physically realistic lines, a gap exists between real urban variation and existing tools**.** Additionally tuning these tools to a visual style can be challenging. The purpose of this project is to introduce stylistic blending to Procedural Architectural Modeling through an existing optimization system in order to generate realistically varied architecture. This tool will generate the layout or arrangement of elements separate from the actual geometry, simplifying the process of adjusting the visual style.

Project Blog: http://shillsarchitecture.blogspot.com/

1. INTRODUCTION

Whether generating the environment of an open world game or a wide shot in a movie, hand-crafting every building takes far too long. Most cities have been built with a variety of styles, across multiple time periods, in wealth and in poverty. Even the function of buildings change, for example, when old, industrial buildings are converted into loft apartments. Moreover, each building's geometry must be generated so as to be both believable and internally consistent. Consider Philadelphia: the Rittenhouse square neighborhood alone hosts a collection of modern towers, converted hotels, row houses which vary from Victorian to art-deco, with the occasional colonial or mission style home thrown in. To the south, the row houses get shorter and less maintained and decorative. To the west, former industrial warehouses line the river and to the north, international style skyscrapers cluster around City Hall.

Without serious stylization, a city cannot be populated from a single template-- cities have variety, which, when lacking, is glaringly obvious to the observer. At the same time, streets, districts, and neighborhoods are often dominated by a single aesthetic. To create realistic, visible transitions from one urban district to the next requires stylistic blending. Each building must be able to blend at least two styles, with aesthetic discretization to create visually comfortable transitions.

This paper proposes a system to describe building styles and the constraints of a particular site for an existing optimization solver. The program will create a script or file export that, when solved optimally, can be interpreted geometrically. Even after optimization of the layout of architectural elements, it should be possible to edit decorative subtleties with out re-computing. For example, switching which moulding profile to use for a cornice from the program's random selection, to one of the user's choice. Style blending may also be editable separate from the facade layout, if the architectural elements being used have sufficiently similar proportions.

This project makes the following contributions:

- Creates an extensible system for generating procedural facades and roofs.
- Allows easier definition of additional styles
- Allows blending of styles

1.1 Design Goals

Under some circumstances, artists and/or production companies (movies or games) lack the modeling manpower to create fully detailed assets as background content for cities, especially in wide or aerial shots. This tool will allow users to generate buildings with controllable amounts of variation on stylistic blending, to create detailed, realistic and/or stylized cityscapes. The output will be in a standard, widely accepted format.

1.2 Projects Proposed Features and Functionality

- Optimization based arrangement of elements with constraint input for the individual building facades and roof lines
- Blending between architectural styles in both layout and visual styling
- Randomization of unconstrained values to create variety

2. RELATED WORK

In order to begin building generation, a user must consider the architectural style they will be working in, which defines the proportions and shapes (e.g. classical); the visual style they will create, which determines the tone of the building (e.g. ornate or simplistic); and the realities of a building (e.g. overal dimensions, full building vs facade). Procedural Architectural Modeling has gotten scholarly and industry attention recently, yet the industry still seems to lack artist-friendly, versatile tools that allow the user to usefully consider all three elements. Thus far, work on Procedural Architectural Modeling has broken down along the following lines.

I. Versatile, but difficult to style

Tools like CityEngine exist for the technical users, but conforming buildings to a particular visual style remains difficult, even once a user is able to capture an architectural style. Further, CityEngine is based in formal grammars, which makes editing the styles difficult for users unfamiliar with them. To compensate, CityEngine is a very flexible system, capable of creating almost any architectural form, if the user puts enough effort into learning the rules and fine-tuning the visual style. This software has set the bar for versatility and interactivity very high. Therefore, this project will add stylistic blending as a new feature, rather than improve the existing features significantly.

II. Inflexible, but stylistic

On the other side of the coin, tools exist to generate very specific kinds of architecture easily and effectively. For example, consider the description of Frank Lloyd Wright's prairie houses as a formal grammar, but as one that needs no further adjusting [KE81]. This simple tool will quickly and easily generate a very specific style of building, having effectively traded versatility for ease. Compared to these, the proposed tool should be far more flexible, though not as physically grounded or as finely tuned to a single style. Similarly, work on recreating existing buildings from photographs has been able to generate architecturally realistic, stylized buildings, but it is perhaps the least flexible of the systems being discussed [MZW*07]. This approach gets realism for free by using actual physical structures as its basis, but in doing so, it can only recreate what already exists, not create anything new. Theoretically, one might feed the system images from different buildings as if they were from just one, but it is unlikely that the results would be useful or predictable.

II. Realistic, but inflexible

Physical stability modeling, approaches Procedural Architectural Modeling along realistic lines. With a blend of grammatical rules and optimization approaches, the tool first creates structurally sound masonry buildings that follow grammar for the placement of elements, and then optimizes forces from the physical model to adjust the size and proportions of those elements [WOD09]. This approach produces wonderful results, but is clearly unhelpful for the timber frame or concrete slab construction common to more recent construction. Realistic optimization has also been turned to floor-plan creation, which uses graph representations of room connectivity and rough constraints on size [MSK10]. The results produce wholly reasonable suburban homes in a variety of styles, but again lacks the flexibility of producing a wide range of architectural or visual styles. While there is some maneuverability within the scope of the program, there is little room for design or imagination.

Thus far, the ideas laid out by Wonka and Kelly [KW11] in their paper "Interactive architectural modeling with procedural extrusions" represent the closest scholars have come to realizing the ideal of both versatility and stylistic control and is some of most advanced work to date on procedural architecture. Their discussion of stretching elements and creating complicated roof lines is highly instructional, and easily adapts to architectural and visual styles, although their extrusion-based approach has not, thus far, been used for refined detail or for stylistic blending.

3. PROJECT PROPOSAL

The tool proposed will expand on existing work with the addition of stylistic blending, in order to generate greater architectural variation while maintaining visual detail.

3.1 Anticipated Approach

By using an external optimization system, the first feature of the proposed tool will be the description of architectural styles and accompanying geometry as constraints. I will describe rules for symmetry, repetition, spacing, nonuniform scaling, roof pitch, and other spatial constraints for use in the optimization phase. Spacial and proportion rules for different styles will vary numerically, allowing for interpolation to obtain intermediate values. I can then further blur the distinction between styles by creating geometry to match a few discrete steps of the transition. As mentioned, variation within the same set of constraints can be achieved by randomizing unconstrained values. Further variance can be achieved by modeling and randomly selecting variants of individual elements within a single style or transitional step. Future users should be able to create an element set and apply existing rules to create buildings with similar proportions, or to adjust rules using old elements to create new buildings in old styles. The work flow should function as follows: 1) choose and/or modify a set of constraints, either in a text file or possibly with a simple UI; 2) select a set of elements from a library to be used in visualization; 3) run the optimization through AMPL for a desired number of iterations; and 4) take the result from AMPL, position and adjust the selected elements accordingly, and display the geometry in a GL window. If the user is satisfied, the result can be saved to an OBJ file. If the user is not satisfied, they can revise the original result by swapping elements for variants or entire sets, or simply by altering the constraints and running AMPL again for an iterative process. Swapping entire elements sets would likely cause the layout to be less optimal than originally generated, but the result could still be acceptable, depending on the user's design goals.

3.2 Target Platforms

This project will make use c++ and/or Python to interface with AMPL and Snopt, then visualize the final geometry in GL.

3.3 Evaluation Criteria

The completed tool should allow the user to freely create buildings that fit spatial constraints in a pure or interpolated style. The geometry and appearance of the buildings should be pleasing and hold up to other works in terms of plausibility (where intended) and architectural consistency (where intended). Stylistic blending should provide new ways to adjust the appearance of generated architecture.

4. RESEARCH TIMELINE

Project Milestone Report (Alpha Version)

- Completed all background reading
- AMPLE scripts generates simple test building
- Test geometry tagged and functional for viewing
- Final geometry outputs directly to file
- Tentative input UI

Project Final Deliverables

- Input mechanism fully functional
- Two fully realized styles in constraints and geometry
- Functional blending
- Optimized façade layout generation
- Post optimization and iterative editing
- Demo buildings and interpolation sequences
- Documentation

Project Future Tasks

- Batch generation for multiple buildings at once
- Use of footprint polygons or splines for freestanding buildings
- City generation with two dimensional interpolation of styles across area
- Better support for free standing buildings
- Support for abstract architecture (e.g. Ghery)

5. Method

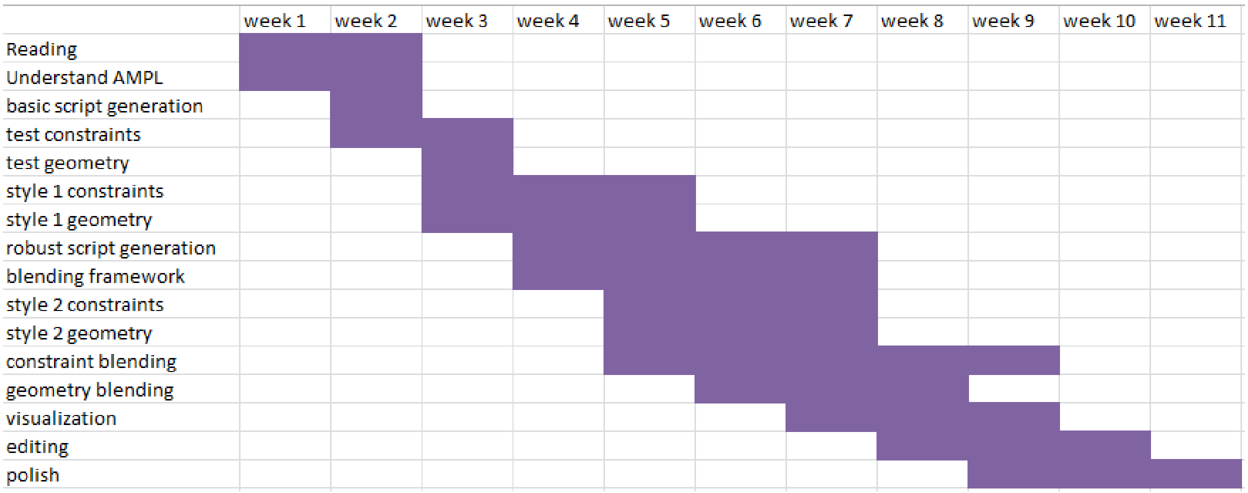
6. RESULTS

7. CONCLUSIONS and FUTURE WORK

References

[COR86]	Le Corbusier. 1986. *Towards a New Architecture*. Trans. Frederick Etchells. 3rd Edition. Dover Publications, Inc, New York, NY.
[HJ66]	Hitchcock, Henry-Russell and Philip Johnson. 1966. *The International Style.* W.W. Norton & Company, New York, NY.
[KE81]	Koning, H., and Eizenberg, J. 1981. The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B.* 8, 3, p. 295-323, 29 pages. http://dcom.arch.gatech.edu/ellen/secured/flw-sg.pdf
[KW11]	Kelly, T. and Wonka, P. 2011. Interactive architectural modeling with procedural extrusionz. *ACM Trans. Graph.* 30, 2, Article 14 (April 2011), 15 pages. DOI = 10.1145/1944846.1944854 http://doi.acm.org/10.1145/1944846.1944854
[LWW08]	Lipp, M., Wonka, P., Wimmer, M. 2008. Interactive Visual Editing of Grammars for Procedural Architecture. *ACM Trans. Graph.* 27, 3, Article 102 (August 2008), 10 pages. DOI = 10.1145/1360612.1360701 http://doi.acm.org/10.45/1360612.1360701
[MCP02]	Michalek, J. Choudhary, R., Papalambros, P. 2001. Architectural Layout Design Optimization. *Engineering Optimization.* 34, 5, p. 461-484, 24 pages. http://gdi.ce.cmu.edu/docs/architectural-layout.pdf
[MSK10]	Merrell, P., Schkufza, E. and Koltun, V. 2010. Computer-generated residential building layouts. *ACM Trans. Graph.* 29, 6, Article 181 (December 2010), 12 pages. DOI=10.1145/1882261.1866203 http://doi.acm.org/10.1145/1882261.1866203
[MZW*07]	Pascal Müller, P., Zeng, G., Wonka, P., and Van Gool, L. 2007. Image-based procedural modeling of facades. *ACM Trans. Graph.* 26, 3, Article 85 (July 2007). DOI=10.1145/1276377.1276484 http://doi.acm.org/10.1145/1276377.1276484
[SUM63]	Summerson, John. *The Classical Language of Architecture*. Cambridge: The MIT Press; 1963.
[WOD09]	Whiting, E., Ochsendorf, J., and Durand, F. 2009. Procedural modeling of structurally-sound masonry buildings. *ACM*

	Trans. Graph. 28, 5, Article 112 (December 2009), 9 pages. DOI=10.1145/1618452.1618458 http://doi.acm.org/10.1145/1618452.1618458
[VIT60]	Vitruvius. 1960. *The Ten Books on Architecture*. Trans. Morris Hicky Morgan. Dover Publications, Inc, New York, NY.



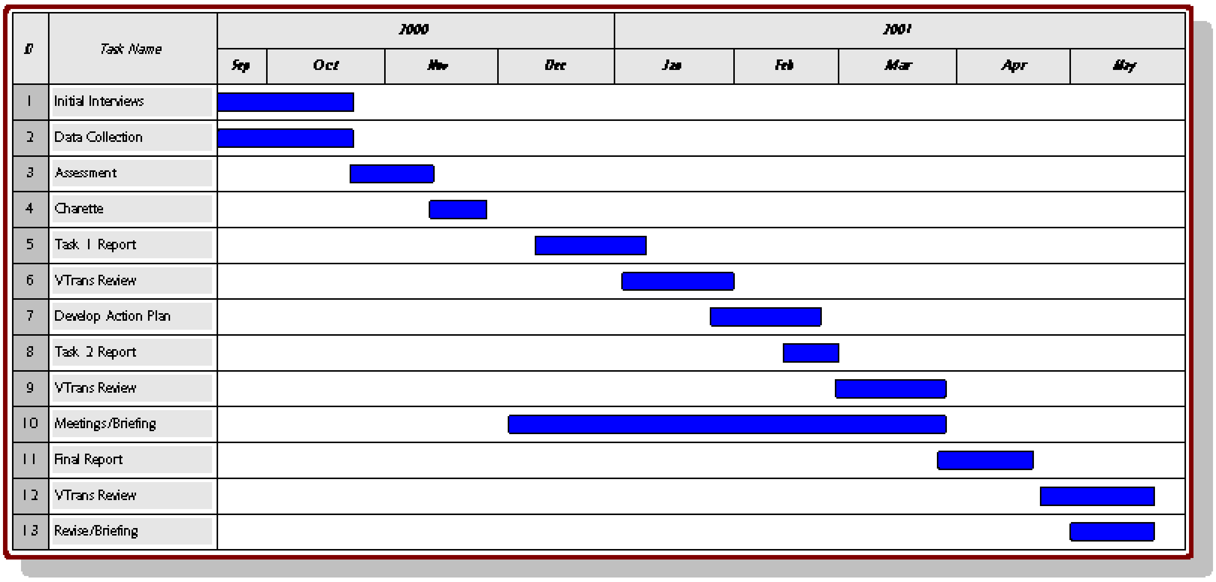


Figure 2: *For publications with color tables and figures that span two columns like your gant chart or results will.*