SCRIPTING ANIMATION

Toward the Capture of Computational Topologies and Articulation of Change

CARL R. LOSTRITTO

University of Maryland
carllos@umd.edu

Abstract. Relationships are amplified and collapsed together as animate surfaces in the formal and spatial manifestation of design parameters. Animation is demonstrated as a medium to express topologies, as each frame is the resultant of a programmed series of computations, the result of which varies with a parameter: effectively, time. Related conditions are parameterized through the design of algorithms as a means of direct translation into animation. Interrelated forces and limits can conversely congeal into statics with animate qualities. Process and product merge to create a language of phenomenological effects and patterns. While animation is exploited to represent parametric relationships there is a maintained awareness of time and space. The systemization and codification of design “problems” simultaneously facilitates functional, solution-driven architecture. The output is characterized by complex, performative, and specific solutions uniquely relevant to emerging models for fabrication and construction. Usage requirements and site conditions carry the weight of information-based contexts and experience-based symbols as fuel for the inherently cyclical process.

Keywords. algorithm: animation; scripting; computation; aesthetics.

1. Background

Mark Goulthorpe writes redundantly but essentially that, “Perhaps it is not stating the obvious to suggest that animation animates(!), produces an effect...” (2001) To be animate is to be considered full of life and vigour, two traits with which architecture is rarely associated. As technology enables form and, more specifically, buildings, to manifest in nearly boundless fashion, animation
research provides the medium for designers to capture and explore previously intangible conditions. This means that animation can expand beyond a tool for simulation or representation. If architecture can be considered animate than the animation must be translatable to architecture.

The process of scripting necessitates both specificity and abstraction. Conditions, motivations must be defined as discrete. Terzidis posits that architecture is historically algorithmic and that the computation, when harnessed at low-levels can be expressive. (2006) Algorithms, if they are to be useful beyond a single end, must be general enough to expand to other contexts. Beyond the goal of efficient production, or the false-rationalization of leveraging parameters as fuel rather limits, algorithm design implies that the algorithm itself is valuable.

This research explores how these two design media can serve to enhance, validate and transcend one another in the production of form and space. Considered in isolation animation and scripting expand the design process, promoting infinite possibilities as flexible armatures for visualization or generation. Their combination here serves to, through the tracing of their processes, become form and space.

2. Animation and algorithm, linked by digital computation

Animation considered independently from video or film is defined as the illusion of motion through the repetition of slightly varying discrete instances. Whether digital or analogue at the source, this effect is achieved through the rapid display of a series of static frames. Animation in the only media suited to directly represent change. The perception of motion and the perception of a series of static instances are mutually exclusive. The latter, however, is the best means to represent the former. The illusion of motion results from an inability to perceive individual frames. As a result of exceeding the rate at which human perception can distinguish between frames, continuous change is perceived. Animation can be thought of as the representation of a condition of change at the sacrifice of understanding of discrete instance.

Scripting is considered here as the execution of algorithms within the framework of existing types and methods within software environments. The narrow technical definition of this term need not necessarily exclude the reference to dramatic contexts (as in the “script” of a play). Scripted code and scripted dialogue both work within an established language but, unlike architectural drawings, don’t strive to prescribe or even predict the outcome of their manifestation with, respectively, variables or actors. With the historic and conceptual value of the algorithm established, a primary concern becomes the traceable affect algorithm can have on output.
A link is established between scripting and animation when formal/spatial conditions are algorithmically generated to vary with a parameter. When an algorithm is placed within an armature of animation and that parameter is valued as the current time in the played animation (or the value of the current frame within the sequence of frames), the outputted movie is the articulation of the algorithm itself. Each frame is the output of one complete execution of the script; the exact result is dependant, to some degree, on the current time. The same script is executed n times (where n is the number of frames) to generate an animation. As the discrete value T, time, updates with each subsequent frame, the results from that particular execution of that particular frame vary (sometimes only slightly) from the previous iteration. Because T is, by definition a varying parameter, results are varied but consistent. The animation then, viewed as a singular event, portrays, through motion, the relationships embedded in that algorithm.

Figure 1 reflects one such expression where the volumetric limits of a packing algorithm very directly. The resultant packed forms vary indirectly, as their position is the result of a non-linear and algorithm. In this process the nature of the packing itself is valued and expressed over any one packed instance.

![Figure 1](image)

*Figure 1. Animating an algorithm that packs, orients and aligns program volumes within a boundary that varies with “time.”*

A cyclical process of abstraction, articulation and testing, empowers a computational design methodology with the potential for spontaneity, surprise and discovery. It is through these means that programming, scripting, and algorithm design generally can be applied has more than problem-solving tools. Rather, they are mediums by which to explore, design and create. The reverse linkage is available between motion graphics and film as art form and the relationship-laden reality of built architecture. Film and motion graphics can therefore be used as precedent and generator where the conventional role is limited to that of inspiration.
3. Reinventing digital animation from the algorithm up

As much as scripting an animation are conceptually related, even interdependent, their technical merging requires a low-level approach.

In 1877 Edward Muybridge used animation as an analytic device toward the settling of the “galloping question.” Muybridge proved that indeed there was duration of time during the gallop of a horse in which all four of its hooves did not touch ground. (Clegg, 2007) The method was simple, known motion was parsed into frames though what at the time was considered rapid succession photography (today, “film”). Viewing a single frame in isolation was proof. This research seeks also to use animation analytically but in the reverse fashion. Animation is constructed by a computationally controlled generation of individual, sequential frames.

Many software applications offer powerful environments for conveniently generating animations. However, these applications operate through the controlled transformation of existing conditions (the moving a camera, scaling an object, undulating a surface, as examples). Animation produced this way can be viscerally compelling but is often specific and, as a result, limiting within a design process. This is also far from ideal intellectually. As Terzidis explains, “The dominant mode of utilizing computers in architecture today is that of computerization entities or processes that are already conceptualized in the designer’s mind are entered, manipulated, or stored on a computer system. In contrast, computation or computing, as a computer-based design tool, is generally limited...some venture into manipulations or criticisms of computer models as if they were products of computation...mouse-based manipulations of 3D computer models are not necessarily acts of computation” (2006). If animation is to occur anywhere other than the conclusion of the design it must be translatable to other media.

When an animation is generated directly from a controlled algorithmic process driven by parameters (as opposed to the typical animation toolsets embedded within software that serve to articulate form and simulate experience) any frame captures a solution to the designed problem. By visually capturing an algorithm, complete with conditionals, operations and parametric neutrality, the animation reveals topologies, geometric—and potentially spatial—relationships emerge.

An example of such a generative and conditional condition occurs when spaces are subtracted from volumes—a Boolean, binary and conditional operation, as in Figure 2. The resulting form has no predetermined geometric typography and in some frames may not exist at all.
Figure 2. Animating an algorithm that packs, orients and aligns program volumes within a boundary that varies with “time.”

An animate armature develops in which the conventional programming structure emerges through the use of effectively global variables, referenced and modified within each frame as animations capture the interaction of multiple functions eventually towards the conception of built form.

4. Societal context for phenomenal effect

Algorithms in raw form are neither complex nor complicated but can produce an outcome richly layered with an ambiguous combination of complexity and simplicity, unflatteringly termed “simplicity” by Kaijima and Panagiotis (2008) in their derision of “computational decoration.”

Conversely, absurdly complicated code is often necessary to replicate the relatively instantaneous but non-algorithmic heuristic approaches by which the human brain operates.

The output presented in this research utilizes a real organization as hypothetic program. The United States’ National Aeronautics and Space Administration headquarters in Washington, DC presents the necessity to explore the articulation of a structured design process with a complex but ordered aesthetics. At NASA, rationalization, systemization, problem solving, and logic are a way of life. NASA’s public face reminds society that science and facts represent our most profound hope toward an expanding sense of self, society and universe. Figures 3 and 4 document resultant outputted forms, mutually influential and dynamically performative. Further though, these forms imply the algorithmic and animate processes that served as their genesis. Computational relationships, are manifest as topologies in the form and the spatial experience.
5. Interpretation through the lenses of perception and convention

The process of scripting necessitates that some functional requirements be satisfied directly. It is important then, to regularly evaluate the output using additional standards. Orthographic drawings and simulated experiential graphics
promote an engagement with the architecture through conventional standards. Besides serving as means of standardization, these drawing methods compress and unify through a process of dimensional isolation. Orthographic drawings articulate of topological aesthetics in an image.

It is also reasonable to conclude that to some degree the resultant architectural form is inconsequential and ultimately irrelevant given that this type of process – computational, temporal design – inherently results in a set of solutions rather than a single ultimate solution. Specific formal and spatial effects achieved directly and indirectly from the specific design process can be examined through a perceptual lens to explore. It is not necessary to read, study, and understand every detail, or any detail potential, of the logistics of the computational design media. Instead, algorithms and animations are manifest directly as image. These documents are themselves scripted artefacts, simultaneously reflecting early experiments in the meaning of media and representing a potential architectural outcome.

8. Conclusion

This design research has demonstrated that spatial and aesthetic effects can be meaningfully synthesized within the architectural design process. The outcome is more than a digital, computerized, or computational aesthetic. Instead, the result is output and description of the design process simultaneously. This process has established the primacy of topology in the maintenance of relationships through the translation between media, scales, and perception. To that end, complexity emerges from the efficient solving of problems and in the creation of discrete relationships.

Restrictions, parameters, and rules can be embraced simultaneously. The abstraction that fuels the designer must exist through representation and synthesis rather than simplification pre-patterning.
References


