Computation as an Ideological Practice

NATHANIEL ZUELZKE
École Polytechnique Fédérale de Lausanne

TREVOR PATT
École Polytechnique Fédérale de Lausanne

JEFFREY HUANG
École Polytechnique Fédérale de Lausanne

Computation is not simply a technical question but an ideological one. Unlike automation, optimization, or even computerization[1], in which predefined routines process input and return predictable results with little intervention necessary or possible by the user, computation does not require the relinquishment of the author's capability to act.[2] Regardless of the tools and media used, a computational process demands the explicit identification of inputs, formalization of behaviors, and integration of these relationships into a cohesive, interdependent system.[3] This pervasive scope and profound depth of necessary decision-making means that a computational procedure embodies the beliefs and values of its author — even when this embodiment is not recognized. Computation supports an infinite range of underlying values, and these values may be expressed in any number of ways. So while computation is inherently mute regarding style[4], it is a strong assertion of agency and an author's ability to determine a solution to a problem. "Becoming computational" involves acknowledging this agency and assessing its impact in the design process.

The design studio presented here covers two semesters of research into the development of an academic campus and "knowledge city" masterplan anchored by an extension of the École Polytechnique Fédérale de Lausanne in Ras al Khaimah, dubbed EPFL-Middle East. Conducted in the 2010-2011 academic year by EPFL's Media x Design Lab[5], the design brief presents several timely provocations in sharp relief. In the wake of the global economic downturn, the plausibility of much ongoing development is uncertain[6] demanding more adaptable development over time. The global warming controversy has engendered a cultural mandate for sustainable environmental responses. Population growth and rapid urbanization in developing areas demand new protocols for settlement. Situated in an isolated and changeable desert context, the EPFL-ME project must respond to these imperatives simultaneously and synthetically.

The multiplicity of these design provocations and their likelihood to shift over time suggests that solutions to the brief would not have a unilateral or static set of design strategies guiding development of the campus. The vastness of the site and desire for environmentally-responsible architectural responses implies the consideration of formidable quantities of geotechnical and meteorological data. Further, the unique environment and cultural context in the Middle East lead us to eschew Western notions of what a campus ought to be, seeking instead distinctly situated proposals without necessary reference to existing typologies or precedents. In order to achieve this temporality, complexity, and novelty, each semester focused on the computational development of a three-dimensional digital model.

Students worked individually or in pairs to define and refine a single synthetic tool generated from an analysis of the site's environmental properties over time, incorporating an increasing number of parametrically-defined design considerations, and yielding a mathematically-precise geometrical representation of an architectural solution which could be evaluated according to the logics of its formation. Because of this dual role as both data-driven formal generator and integrated evaluation mechanism, we term these model-tools "computational engines". In this pervasive role, computation is simultaneously a technique and a deep structure, enabled by technology and organizing the entire design process from conceptual genesis to precise description. Engines created in the first semester were concerned primarily with large-scale planning logics and formed the basis for second semester research concentrated at the mezzo-scale. As such, computational engines are seen as extendable to other design problems and through additional stages of realization which were not within the scope of this research.

Rather than a transformation of conventional pedagogical instruments, this course was conceived as a revision thereof. The architectural studio is a powerful and yet-viable educational methodology in need of adaptation for digital media and technologies.[7] The incorporation of computation's ideological practice into the studio's methods of "learning by making" achieves such a reboot, intensifying and structuring design exploration.

CLARIFYING "COMPUTATION"

For the sake of precision in terminology, it should not be assumed that all computer-calculated models are "computational" as the designation is intended in our context. Computational design connotes a moot position, yet this position remains ambiguously defined and often misunderstood.[8] It is readily invoked by formalists and rationalists alike, who in the extreme either disown authorial agency[9] or naïvely claim objective optimization.[10] It seems to be agreed that computation in architecture is used to produce *models*, which may or may not be adaptable, relational, multiple, or optimizable.

The conventional understanding of an architectural model implies a static instance, corresponding to the mechanical paradigm of identical copies[11]. The scale model represents a single, unique solution to an architectural problem and is intended to be actualized with a high-degree of fidelity, any variation occurring in scalar translation deriving from ambiguity or lack of resolution in the model rather than intended customizability. While most closely associated with analog representations, static digital models also belong to this understanding.

Interpreted as an exemplar, an architectural model may also be understood as an archetype, abstracted and idealized in its definition but intended to be adapted when physicalized. Predating their contemporary digital counterparts, parametric patterns have been used since Classical times to describe proportional systems and narrative procedures for making.[12] While parametric systems assert mathematical relationships, they need not be calculated by a computer. However, we will use "parametric" to denote necessarily digital and procedural techniques which embody a family of related solutions. Animate models created with the aid of calculus-based solvers are dynamic and evolutionary[13], and tend to share formal characteristics often present in computational models. However, the mediation of internal considerations and external forces evoked by key-frame animation softwares' continuously heterogeneous geometries is largely representational, having been produced by interpolation between static, determined states. Similarly, the variable resolution of "versioning"[14] is a multiplication of instances rather than a true subset of a rigorously defined and structured solution space.

Fundamentally, computational models are parametric simulations which extend the capabilities of the latter two dynamic types of models. They are parametric in that their relational constraints can be varied to produce multiple related solutions quickly. They are simulations in that their constitution is procedural, dynamic, and inherently temporal. As such, computational models have the capacity to be used as design engines, hierarchically structured and representative of a unique subjective agenda, which can resolve conflicting objectives and provide real-time feedback in the design process.

The creation of a computational engine was paramount within the studio's methods. Students began

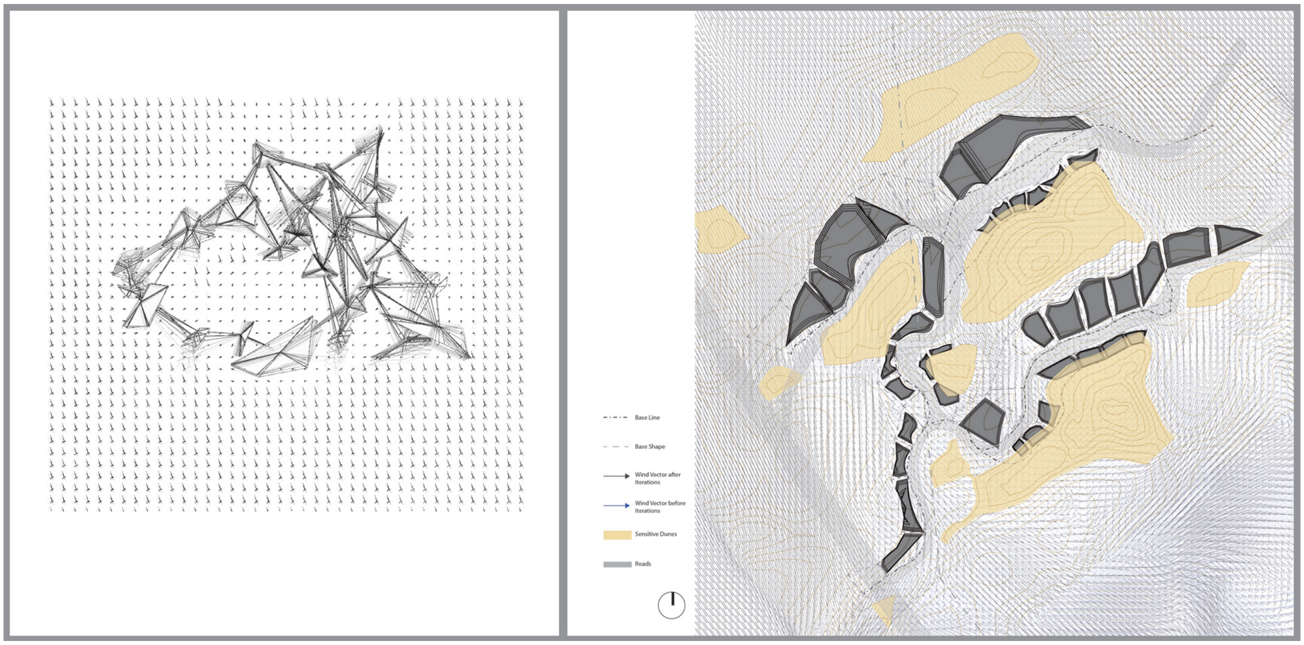

Figure 1. Solar exposure as a function for identifying landscape features

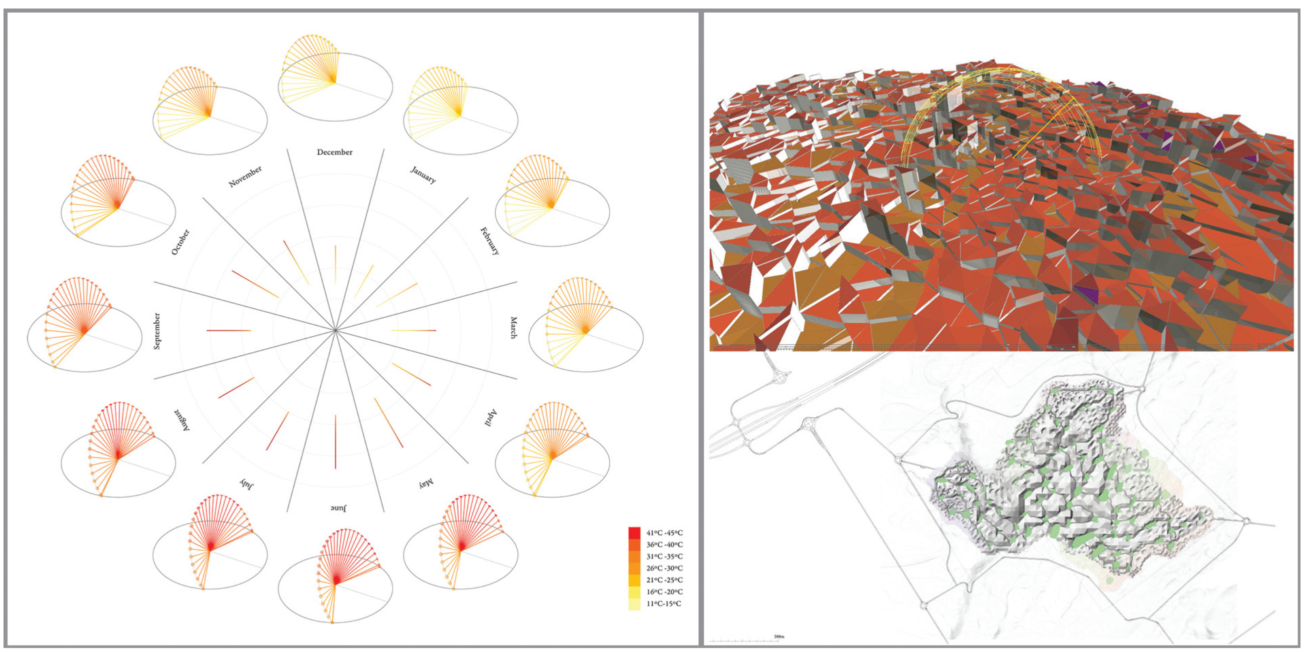

Figure 2. Simulation of dune movement and application in a fluid-dynamic reactive masterplan

by analyzing the site and its topography according to local climate data. The consideration of environmental data in light of the site's physical, geometrical properties revealed zones of differentiation and led each student group to identify of a unique design agenda. Formalized objectives which support these agendas such as circulation concepts, programmatic diagrams, and material constraints were then incorporated into the parametric site maps to physicalize, ameliorate, or exacerbate the revealed conditions. This incorporation occurred iteratively using the engine-in-process to explore its embodied solution space quickly and providing feedback for refinement. Ultimately, the computational engines were presented by the students both as a document of their architectural proposals and as a visual argument supporting the proposal and demonstrating its logic.

SITE AND SITUATION

The parameterization of a site requires a radical rethinking of the initial conditions of design. In a typical project these might be some combination of building codes and zoning regulations, typology, and program. Increasingly, architects are finding themselves called upon to design in locations or situations where these guidelines do not yet exist. New cities constructed at rapid pace, cities of unprecedented height and sprawl, and cities with new relationships to exterior or public space are being produced across the globe.[15] What we find more often than not is a relocation of the same typical guidelines copied intact from the historic Western city or its suburbs into new, foreign surroundings and pressed into roles they may not be able to fill. On the other hand, the parametric site analysis begins not from formalized laws or idealizations, but, like a scientist translating an experiment from the lab into the field, realizes that the disruptions of existing conditions are immensely significant to the successful implementation of any plan. The model must be able, at the least, to endure the disrupting influences of the environment; at best it engages with them channeling these external forces into productive, symbiotic responses.

First among the factors restricting development potentials in this project are the challenges of the existing landscape. Nearly the entire site is covered in sand dunes of various sizes. The largest of these reach up to 100 meters in height and create steep, difficult to occupy slopes, but also opportunities to avoid direct sun (Figure 1).

Smaller dunes, on the other hand, while less formidable topographically, are found to move more quickly across the site, up to 60 meters in a year. Evidence of this movement was clearly visible as cleared roads were seen to be drifting closed between our two visits to the site occurring 5 months apart. The movement of the largest dunes was traceable through satellite photographs and was seen to move, on average, 200 meters between 2004 and 2010, a time span roughly equal to the planning and construction timeline of the EPFL-ME campus. The mutability of the landscape means that the physical definition of the site geography was a much more dynamic influence on the planning strategies than other sites might have offered. Modeling

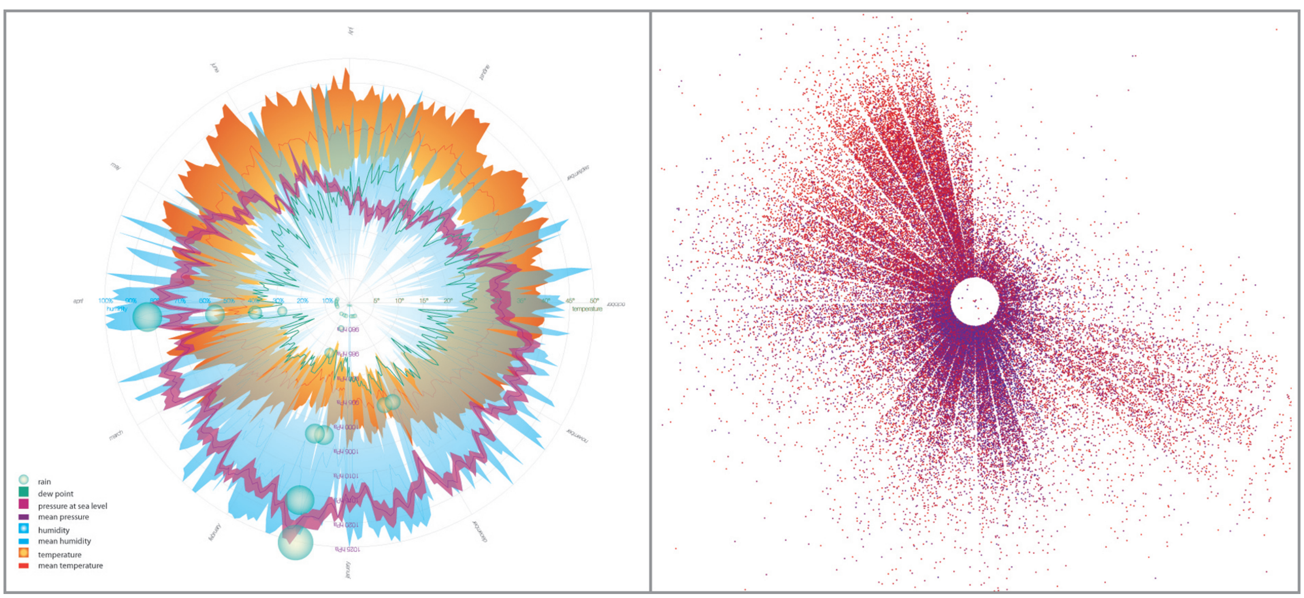

Figure 3. Left: comparative graphing of 2009 hourly weather data—precipitation, air pressure, humidity and temperature. Right: hourly wind data for 2000-2009—direction, speed and temperature.

and measuring this data with computational methods allowed this influence to be incorporated within the design process directly (Figure 2). The movement of dunes points to the second incentive to computation. The climate of the region can only be described as extreme. Temperatures can reach 48°C and rarely dip below 23°C during the daytime. Rain is incredibly scarce, though morning fogs do occur. Wind data shows predictable patterns for the dominant winds on diurnal and seasonal scales, but the effect of these winds on comfort vary widely based on a combination of ambient temperature, humidity, and speed (stronger winds, especially the *Shamal*, a regional wind responsible for the most violent sandstorms, mean airborne sand particles). Working computationally allowed students to measure, analyze, and visualize datasets consisting of hundreds of thousands of items.[16] Parametric models and physical simulations were also created to better understand in what ways these environmental conditions might be localized across the site and how the built environment could encourage more comfortably habitable zones to reduce the reliance on artificially conditioned buildings.

While a masterplan[17] exists for the region surrounding the Ras al Khaimah knowledge city, the adjacent development remains speculative and much of it is indefinitely on hold. Not only is there no urban context, at 25 kilometers from Ras Al-Khaimah and 7 kilometers from the Al Hamra free zone, the site has hardly a trace of any built context whatsoever. This means also a total lack of existing planning guidelines for building envelopes, floor areas, density, setbacks, street dimensions, or any other of the typical urban design aids. With the exception of access points roughly positioned along the major roads at the periphery of the site, the future campus is not constrained by any pre-existing construction. Furthermore, the speculative nature of the planning means that construction speed and investment levels are difficult to predict in advance.

The lack of pre-existing planning strategies allowed new definitions to be created which synthesized characteristics of climate, environment, landscape, and built construction. Parametric tools, and in particular simulation models, provided methods for defining and formalizing a logical set of guidelines capable of responding to the physical, social, and economic demands without the rigid restrictions of singular, formal solutions. The parameterized site reveals potentials and opportunities which arise from the gathering together of disparate datasets while also allowing interactive modification of design processes as well as the playing out of these processes in time.

As part of a projected 400,000+ square foot cutting-edge international research facility focused on energy, environment, construction, and information technology[18], the campus complex is subject to a battery of internal logics relating to usage, security, materiality, and structure. These internal constraints are frequently at odds with the external climatic considerations. By formalizing desirable relationships as a parametric range of precise geometries and creating a structured but malleable hierarchy between the formalizations, a computational engine permits reprioritization of design considerations fluidly and facilitates the mediation of internal and external concerns. The tool is both a representation of these sliding prioritizations and a means by which to evaluate their propriety.

Lastly, the constitution of a computational engine is not limited to design considerations that are evidently quantifiable. For an institution seeking to create a unique and attractive brand, issues of identity, ideology, and experiential quality are eminently considered by incorporating precisely formalized immaterial, intangible, or phenomenal concerns alongside of less subjective considerations.[19] This raises a critical point, however. That is, as a system of logic the quality of a computational engine is still contingent upon the strength of its assumptions and the integrity of its definitions, neither of which a computer can generate. Used as a feedback mechanism, however, the computer can serve to inform and verify those assumptions and their formalizations.

Site parameterization is not simply a methodological change, but is characterized by entirely different traits than conventional site analysis. By leaving behind static, predefined planning restrictions, for new, synthetic and relational analyses, an understanding of the site is constructed which possesses a stronger agenda.[20] In this way, the parametric engine assumes an agency which is all too often left in the hands of developers or planning documents long out of date. Since the parameters of analysis are not only relational but explicitly defined as processes this document is inherently dynamic and variable. Through variation and adaptation, the logics of the masterplan, the interrelation of components, and the thresholds at which conditions noticeably shift are rendered more apparent and understandable. As a way to make the significance of planning decisions more transparent this improves on the conventional model which loses cohesion as aspects of the plan are altered over time because its many component regulations are not integrally linked. In contrast, the unity of a parametric plan is not degraded by the introduction of compromises or exceptions.

Not only is the parametric site more flexible and reactive but it allows projections to be carried out into the future by reincorporating the results back again into the analysis. This additive loop of feedback multiplies the number of potential outcomes rather than compressing them. More importantly, it gives insight on the impact of a proposal immediately rather than in separated phases.[21] Conventional masterplans which propose either a single static "end" condition or several landmark phases en route to a fixed

Figure 4. A "constructed" site derived from environmental analysis and resolving potentially conflicting influences.

solution are incapable of responding to changes in the factors which regulate development over time. Frozen in an outmoded context, such strategies deliver ineffective planning in the short-term and require extensive recapitulation in the long-term.[22] Uncertainty in the scope and pace of development is typical in any architectural project, however temporal considerations were particularly relevant for the EPFL-ME project though no more predictable for their pertinence. Because computational engines are derived from dynamic site analyses over time, they are inherently temporal. The formalizations of design considerations are analogous to growth processes allowing the specific temporal response to increase or decrease development according to what the changing context can sustain.

Ultimately it is this collapsing of scales, domains, and phases of design into an integrated self-informing model which is the most unique promise of computation. As a procedural method the parametric model represents a formal logic which governs the formation of a category or type. However the specificity and subtle adaptation of that type into a concrete realization posits a unique solution.[23] The design of a parametric model becomes the construction of an analytical tool which is *simultaneously both* the diagram of a procedural method and the quantitative record of an individual instance.[24]

SOLUTION SPACE

Perhaps the best way to grasp the benefit of computational workflows over conventional ones is in terms of solution space. If we consider the set of all possible solutions to a given design problem, the set lacks innate structure. Conventional design processes begin to narrow this space implicitly with objectives derived using architecture's historical apparatus of deductive problem solving (including but not limited to diagrams, orthographic projections, perspectival drawings, and scale models) or imposed arbitrarily by sources external to the architect (such as clients, regulatory boards, or in the case of the academy, instructors and the curriculum). An individual solution is seized upon for consideration – the criteria for which remain allusive, though they may be communicated as authorial desires. Aspects of the solution may be changed to ameliorate inadequacies or, if deemed wholly insufficient, an entirely new solution is consulted. This method is essentially picking a point within the solution space with little discrimination and then testing proximal solutions for what feels right intuitively. Solutions unknown at the outset of the process may be reached through refinement (again utilizing architecture's conventional instruments), or critique and collaboration, or even serendipity or genius. However the solution space remains essentially unstructured and only a small handful of solutions may be regarded in earnest.

In contrast, the codification of design objectives necessary to the creation of an engine organizes the solution space and creates relationships between solutions which are similar as defined by the explicit formalization. As the design process progresses, additional formalizations construct a system of iteratively refined logic and ensure a rigorous structure in an increasingly specific solution space. The combinatorial nature of the system produces complexity, even when individual formalizations are simple. The solution space is no longer innately amorphous, but is neither homogeneous nor Cartesian. It is not symmetric, reflexive, nor transitive. Its nature is heterogeneous, non-isotropic, and multi-dimensional. Because of these emergent properties, the solution set is not axiomatically known and must be explored. The structure ensures that any parameterization is inherently multiple, containing a family of solutions and including unanticipated results which challenge preconceived ideas. As such, both the quantity of solutions considered and the likelihood of novelty are greatly increased.

To explore efficiently, metrics of evaluation or performance must be designed into the engine's formalizations. These could be quantities such as desirable ranges of area for programmatic spaces, or other such numeric targets as are often critical in conventional workflows. They could also be more complex evaluations based on simulation, such as environmental or structural analyses. These metrics provide immediate feedback and, because of the engine's contrived structure, allow the engine to navigate automatically to adjacent solutions for comparison. While this process achieves a combinatorial optimization of exactly those criteria which have been codified in the engine, it is also possible to intervene, using the engine to only identify adjacent solutions but making the selection manually. An engine's solution space is fundamentally computerized in the sense that it is derived from a series of calculations. Conceptually its mathematical

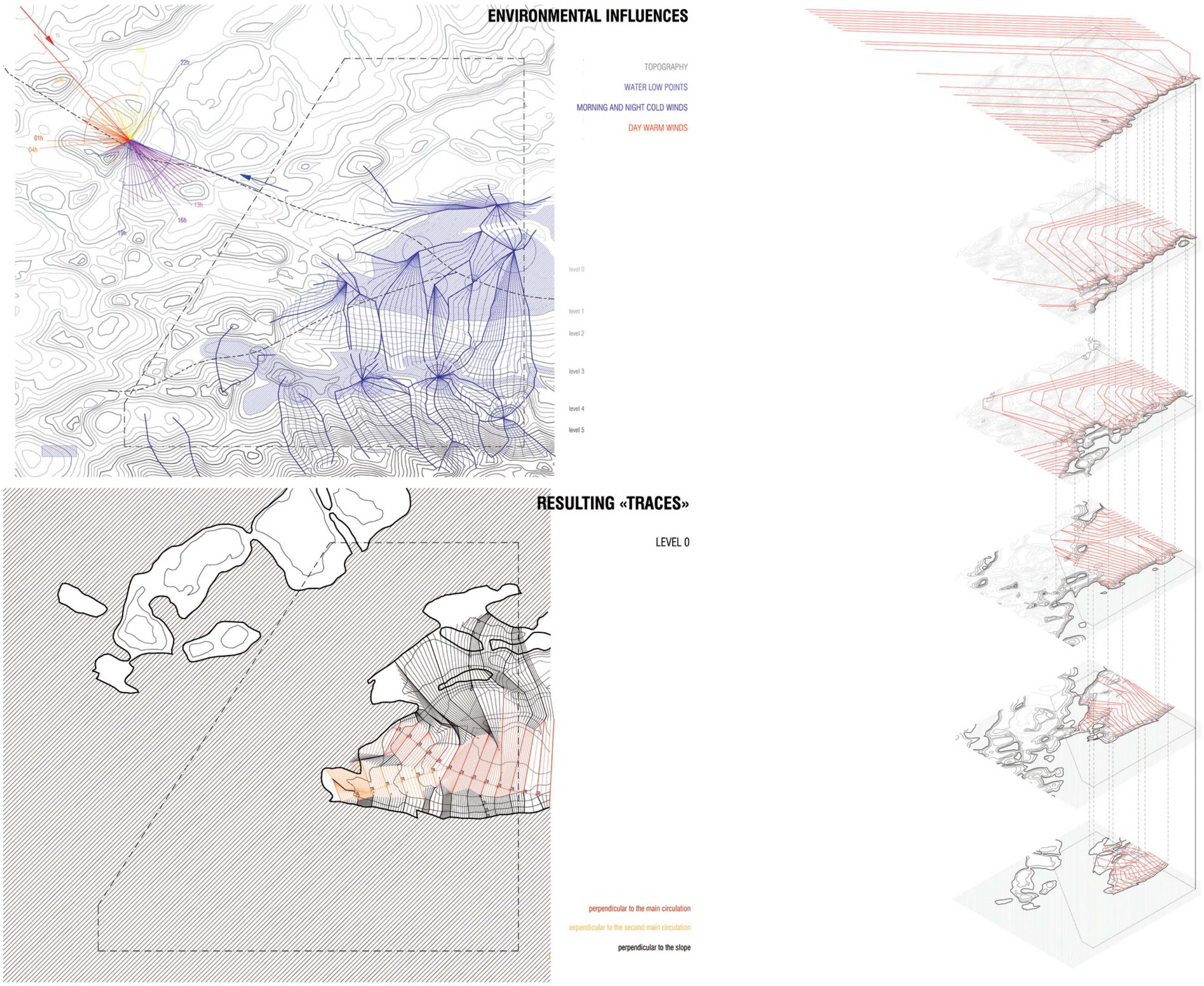

Figure 5. Precise geometric definition of design considerations as a procedure producing unanticipated forms.

description does not require a computer. However, pragmatically such complex description by hand would require prodigious technical faculties and an extraordinary amount of time. The complexity of the solution space is accessible by the computers capacity to perform many calculations quickly. Trial by brute force may be possible in some cases; however, with the feedback of in-built evaluation methods, one may quickly identify desirable subsets of the solution space for further, detailed study and leave less desirable areas unplumbed.

REGARDING AGENCY

It is important to note that a computational engine is not a conclusion but an evolving document which formalizes, refines, and clarifies its authors' intents. Equally important is that this document embodies an agenda, for each of its formalizations – including those of pragmatic or objective criteria – has been subjectively defined by an author.

An engine is an iterative tool which provides visual feedback and real-time analysis of its own premises. It can resolve geometric rigor and subjective design aims in a combinatorial way without *a priori* formal definition. This potential to create spatial configurations without prior determination of their physical qualities affords just enough dissociation from authorial will that a unique invention can be created, yet it is still upon the authors to recognize their own agency and take responsibility for that invention.

Deployed without consideration, computational design exhibits the danger of becoming closed and self-referential. Venturi's words, originally written of Modernists, become pertinent: "The limitations of the platitudinous architects who invoke integrity, technology, or electronic programming as ends in architecture… suppress those complexities and contradictions inherent in art and experience."[25] The recognition of the diverse spatial models that contribute to the formation of architecture in a contemporary digital culture is essential to avoid the descent of digital production into mere rhetoric. It is crucial to understand that computation is not itself an end in a critical design methodology, and in fact it is technology which allows systems of logic to incorporate contradictory design ambitions into a resolved engine.

In the worst case, parametric space as a closed system ceases to respond to considerations other than those of its own internal logic. This possibility is held at bay through the considered design of the space's logic. More bluntly, it is only a tautological danger. Only systems which do not adequately consider relevant design criteria are vulnerable to having not considered relevant design criteria. Again, it is the responsibility of the author to craft a suitable system of design considerations in which external and internal logics play an appropriate role. This is the case regardless of the instruments engaged in any workflow. The strength of a computational engine is that its evolution serves to inform precisely this crafting of logic through continuous feedback allowing its author or authors to have more control, and explicitly reveals the factors and mechanisms at play in the decision-making process.

CONCLUSION

This form of indirect, mediated, or cooperative authorship translated through digital processes is a basis for a computational ideology's approach to design. The authorial intent of the designer is not tempered by the procedural actions of an algorithm or parametric model. The inherent multiplicity does not dilute, but dilates the field of design possibilities at hand. Architectural design, practiced computationally, possesses a unique temporality which escalates the traditionally iterative process of design by drawing together initial premises, processes, and effects produced simultaneously. Such integrated, self-informing feedback gives the impression of automation, but in fact allows (even sometimes requires) the architect to reexamine, reassemble, and elaborate upon the early assumptions rather than accepting them as values givens.[26]

In practice, architecture's solution space might be most accurately described as conventional, that is, it is dictated by convention. As conventions change in response to technology and techniques, the ideological practice of architecture changes as well. However, this relationship need not be unidirectional. By working with a strong ideological agenda it is possible to manipulate architecture's space of inquiry rather than passively accepting the conventions of the time.

With this in mind, we see computation as both a technique and a culture which is employed to update conventional methods. Unlike non-computational design exploration which happens intuitively and arbitrarily (though not necessarily uninformedly), computation organizes and intensifies exploration with an inherent structure. This structure documents the refinement and clarification of the authors' intent throughout the entire design process, providing a critical feedback mechanism which challenges preconceived ideas and illuminates unexpected solutions. Ultimately, computation should not be viewed as a formal – or even methodological – end in itself, but rather as an ideological practice which engenders criticality and promotes innovation.

ENDNOTES

1 For more on the distinction between computerization and computation, see Kostas Terzidis, *Expressive Form: A Conceptual Approach to Computational Design* (London: Spon Press, 2003), in particular the chapter "Algorithmic Form".

2 For further discussion on interaction with software and the scope of authorship, see Mario Carpo, "Authors, Agents, Agencies, and the Digital Public," in *VISIONS: Catalogue of the 9th Edition of BEYOND MEDIA International Festival for Architecture and Media*, ed. Marco Brizzi and Paola Giaconia (Firenze: Image PUBLISHING, 2009), 71-73.

3 For more on General System Theory see Ludwig von Bertalanffy, *General System Theory: Foundations, Development, Applications* (New York: Braziller, 1993).

4 This should be considered in sharp contrast to "Parametricism", the style which Patrik Schumacher claims "emerges" naturally from exploration undertaken with parametric design tools: "Parametricism: A New Global Style for Architecture and Urban Design," *Architectural Design* 79, no. 4 (2009): 14-23.
For more on the relational nature of parametric tools,

see the section "Clarifying Computation" above. For more on the history of parametric patterns as design tools, see note 12 below.

5 See the Media x Design Lab's website, http://design.epfl.ch, and the Ras Al-Khaimah studio website, http://design.epfl.ch/organicites/2010b.

6 Dubai, as example, has a vacancy rate of 30% for residential properties and 50% for offices. See "Gloomy outlook for property rents and prices in Dubai and Abu Dhabi," Property Wire, http://www.propertywire.com/news/middle-east/dubai-abu-dhabi-property-201106285315.html (accessed September 14, 2011) and "Dubai office vacancy rate set to top 50% - study," Arabian Business, http://www.arabianbusiness.com/dubai-office-vacancy-rate-set-top-50-study-293390.html (accessed September 14, 2011), respectively.

7 Jeffrey Huang, et al., *SUPERSTUDIO* (Swiss Cooperation Project in Architecture research proposal, 2007), 1.

8 Kostas Terzidis, *Algorithmic Architecture* (Amsterdam: Elsevier, 2006).

9 Michael Hensel, Achim Menges, and Michael Weinstock, *Emergent Technologies and Design* (Abingdon: Routledge, 2010).

10 Rivka Oxman and Robert Oxman, eds., *The New Structuralism, Architectural Design* 84, (2010).

11 Mario Carpo, *The Alphabet and the Algorithm* (Cambridge, MA: MIT Press, 2011).

12 Mario Carpo, "Drawing with Numbers: Geometry and Numeracy in Early Modern Architectural Design", *Journal of the Society of Architectural Historians* 62, no. 4 (2003): 448-469.

13 Greg Lynn, *Animate Form* (New York: Princeton Architectural Press, 1999), 9.

14 Ingeborg Rocker, "Versioning: In-forming Architectures," *Architectural Design* 72, (2003): 10-17.

15 *New Architecture in the Emerging World*, ed. Oscar Riera Ojeda (London: Thames & Hudson, 2011).

16 Weather data sourced primarily from Wolfram Alpha, http://www.wolframalpha.com/, between September 2010 and May 2011.

17 Compiled by the Ras Al Khaimah Investment Authority, http://www.rak-ia.com/en/rak-master-plan.aspx (accessed September 14, 2011).

18 Franco Vigliotti, "Dean's message," *EPFL Middle East*, http://www.epfl.ae/content/about-epfl-middle-east/dean-message (accessed September 14, 2011).

19 The groundwork for the idea of an "intention space" containing subjective design considerations is laid out in Nathaniel Zuelzke, "Socio-Spatial Becoming" May 2010 [essay], Lausanne: École Polytechnique Fédérale, 12:

> The iterative blending of the spatial constructs of production, representation, and experience produces something I will refer to as 'intention space'. Intention space is the dynamic solution space to the inherently subjective codification of design considerations as interpreted by one or more authors. By its nature, intention space is organizational and experiential and tends to shrink over time as design decisions become more rigidly defined. Etymologically, 'intention' retains an aspect of strain or stretching. The tension between the components of intention space is equally important as the balance amongst them.

20 Alain Badiou, *Handbook of Inaesthetics*, tr. Alberto Toscano (Stanford, Stanford University, 2005), 130:

> …it is impossible to interrogate the traces of an event except under the hypothesis of an act of naming… The question is no longer that of knowing what has taken place, but rather that of making truth out of an undecidable event.

21 Ian Bogost writes about the significance of digital media wherein "Processes are represented by processes" in *Persuasive Games: The Expressive Power of Videogames* (Cambridge: MIT Press, 2007).

22 Nathaniel Zuelzke, Trevor Patt, and Jeffrey Huang, "Course syllabus," *Organicités Ras Al-Khaimah*, http://design.epfl.ch/organicites/2010b/how/studio-information/course-syllabus (accessed September 14, 2011).

23 C.S. Peirce would characterize the contrast as the difference between a *legisign* and a *sinsign*. Charles S. Peirce, "Logic as Semiotic: The theory of signs," *Philosophical Writings of Peirce*, ed. Justus Buchler (New York: Dover, 2011).

24 Bruno Latour revealed parallels in the effect of digital tools in the social sciences especially the increasing equivalence of entities and populations in a lecture "Navigating through Monads: What the digital does to social theory" (presented at École Polytechnique Fédérale de Lausanne, April 14, 2011).

25 Robert Venturi, *Complexity and Contradiction in Architecture* (New York: Museum of Modern Art, 1966), 14.

26 Bruno Latour, *Reassembling the Social* (Oxford: Oxford University, 2005), 31:

> If a given ensemble simply lies there, then it is invisible and nothing can be said about it. The ensemble generates no trace and thus produces no information whatsoever; if it is visible, then it is being performed and will then generate new and interesting data.