

Baroque Rome as Algorithm: Coding History

FRANCESCO BEDESCHI

University of Arkansas Rome Center

LAURA TERRY

University of Arkansas

WINIFRED E NEWMAN

Clemson University

Using principles of design from the pre-modern architectural orders we challenged students to learn algorithmic thinking using Grasshopper™, a contemporary visual programming language. Addressing the relevance of historical theory in contemporary design was facilitated by digital software. Our approach extended to coding environmental and phenomenal processes at the scale of the city for a building program addressing the representation of ‘Smart City’ principles to Roman residents and visitors. Student teams devised programs for a CityMark or urban intervention focused on increasing public interest and awareness of the city as interconnected systems.

INTRODUCTION

The studio targeted the design principles of pre-modern era Rome and the architectural orders in conjunction with our contemporary use of digital tools. We emphasized algorithmic thinking that uses a methodical approach to problem solving with precisely defined instructions. We gave students the shared design kernform or genetic instructions of the pre-modern architectural orders, the analytic specificity of a present-day software program like Grasshopper, and a set of basic actions determined for the design problem.¹ The architectural orders are systematic instructions with a set of special and normal cases as solutions to a given programmatic problem regardless of context. In the case of the architectural orders, once you select an order – Ionic for example, it provides a procedure for generating the formal and ornamental proportional relationships linked to the tectonics of a given material, typically stone. In turn this produces the overall arrangement of a façade and determines the aesthetic outcome of the formal solution. In effect the architectural orders are algorithms or sets of instructions for producing a building façade similar to computational algorithms that are responsive to differing conditions and improvable. Improvability in this sense applied to architecture is the mechanism for the changing and varied design we see in the historic building texture of Rome from antiquity and again in the Renaissance.

Our systems-orientation to the studio pedagogy extended to the environmental and phenomenal infrastructure of the city using the program brief as an opportunity to

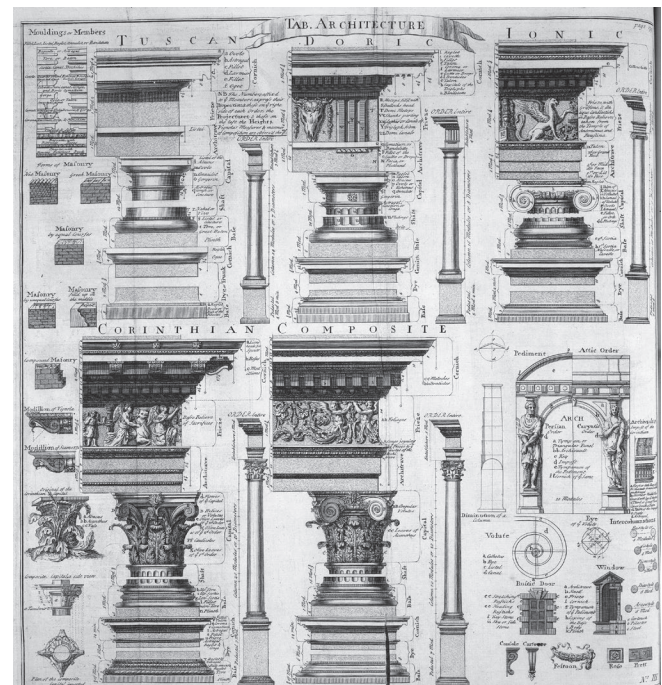


Figure 1. Drawing of the Corinthian Order showing dimensional relationships of the intercolumniation (Source: architectureorders.com).



Figure 2. Map of Via Papalis based on the Grande Pianta di Rome by G.B. Nolli, Rome, Spring 1818 (by author).

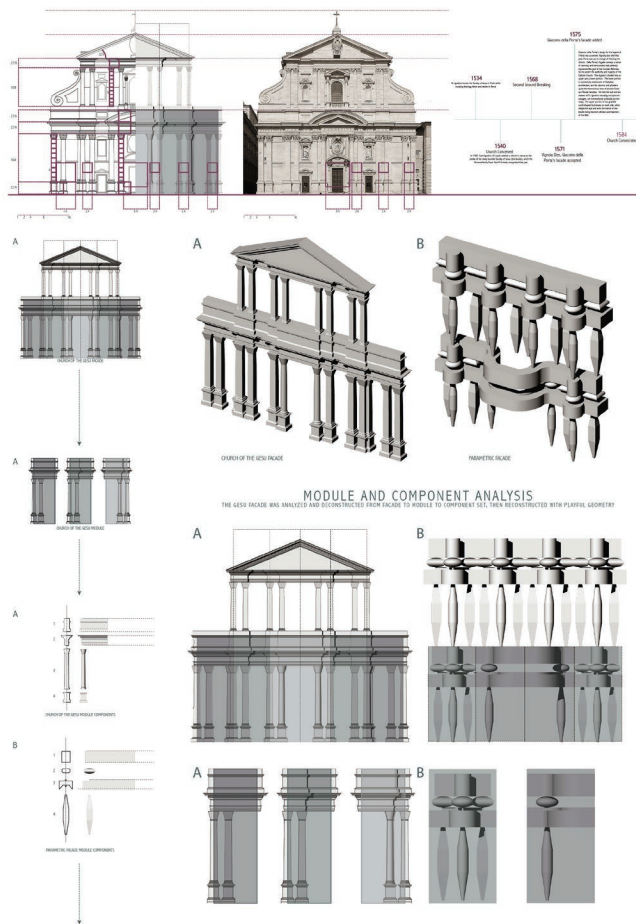
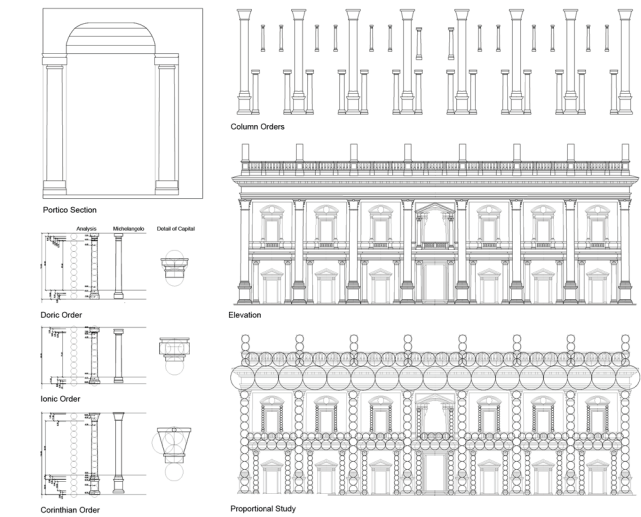


Figure 3. Riley Davis and Peter Slowik, Church of Il Gesù, Rome, Spring 2018. Analysis and Reinterpretation, Rome, Spring 2018. Phase 1 of analysis of the architecture order used in the design of the façade of Il Gesù.

address the representation of ‘Smart City’ principles in Rome. Student teams developed program briefs for a CityMark or urban intervention focused on increasing public interest and awareness of Smart City topics. Six sites were pre-selected and randomly assigned. The studio is part of a study abroad program required of all students and offered in their last three semesters as an advanced options studio. Traditionally, study abroad represents an opportunity to replicate the opportunities offered by the pre-modern École des Beaux-Arts Grand Prix de Rome awarding students a travel semester in Rome, exposure to the Western canon, and a variety of cultural experiences. But globalization makes travel affordable and students today are increasingly aware of foreign cultures. Rather, we approached Rome as a world city addressing the shared problems of adaptive re-use, impact of climate change, and tourism, informed our focus on the Smart City and systems. For us Roman historical architecture continues to play a fundamental role in architecture; more so as we address contemporary issues of parametric design, coding, and algorithmic thinking.



Analysis of Orders of Palazzo Conservatori

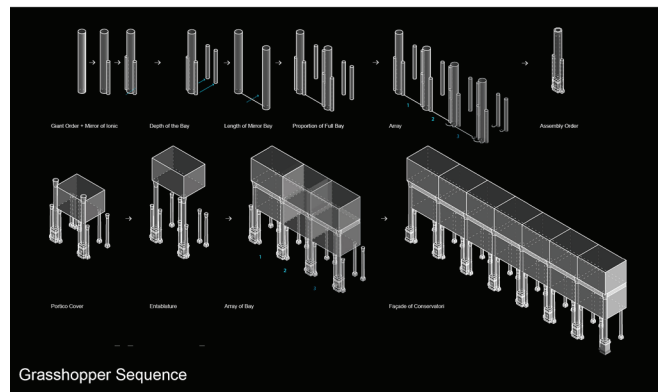


Figure 4. Selwyn Bacchus and Mitchell Willmarth, Palazzo dei Conservatori, Piazza del Campidoglio, Rome, Spring 2018. Phase 1 analysis illustrating the de-coding of the context building and a re-coding using Grasshopper.

KERNFORM, KUNSTFORM, AND CODING

Narrowly considered, it is the ontological status of the relationship between form and matter and its epistemological framework often at stake in specific architecture theory. For this studio we decided to test the limits of German theorist Carl Bötticher’s notion of Kernform and kunstform describing a division between the structural and the ornamental respectively. Bötticher’s theory belies a complexity more apparent in Gottfried Semper’s later equivalents: structural-technical and structural-symbolic, applicable to our contemporary condition of digitally determined form making.² Structure as Semper suggests is the genetic code of the design expressed through materiality, but also contextually defined as a social form. In Semper’s analysis there is an elision between what is visible and present and what is invisible and contingent, similar to the way computational design generates form and matter concurrent with the social and contextual conditions of the autonomous designer. If the architectural orders are both an aesthetic and structural determinant, which they are – they are an ideal place to begin exploring the problematic of an algorithmically generated formal language using digital software.

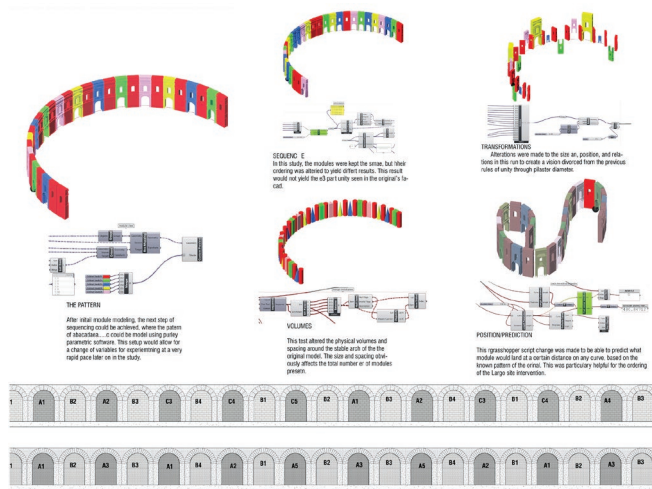


Figure 5. Colt Malloy and Robert Metz, Trajan's Market, Rome, Spring 2018. Phase 1 working code for the semi-circular arcade at Trajan's Market.

The premise of the studio was 'everything old is new again' and the architectural orders used to create Classical and Baroque Rome were revisited in the contemporary context of parametric design. Given architecture theory is a complex of epistemic cultures available to us through a written discourse attempting to explain how or why objects of architecture look as they do, contemporaneous questions such as: What are the conceptual definitions of form in architecture? Is the relationship between form and matter about performance, aesthetics, or both—are especially poignant in the context of historic Rome where the presence of an algorithmic, if not digital, aesthetic is predominant. Even if clearly divorced from the social agenda of a present-day designer, we opted to test one epistemic culture against another to see what would happen.

The Vitruvian Orders and their later interpretation in the Renaissance as the Five Orders are a generative language organized using as base unit of columnar dimensions and their derivatives for all other proportional and scalar relationships. Façades in Renaissance Italian architecture are load bearing stonewalls whose dimensions are given according to the formula for the order – hence the 'code' determines aesthetics and structure (Figure 1). We challenged the students and ourselves to see the contingent and invisible kernform as the 'core-form' or principle informing what is present in the formal language of the architecture of the city and its systems as a way to develop their own 'design code' for the project. The studio was organized into three phases discussed next.

METHODOLOGY: PHASE 1 – CITY (BUILDING) AS CODE

Student teams worked on preselected sites located along the Via Papalis in Rome, a historic processional route from Basilica Papale di San Pietro in Vaticano to the Basilica di San Giovanni in Laterano (Figure 2). Sites were selected for their historic significance and urban location. Phase 1 required students to select

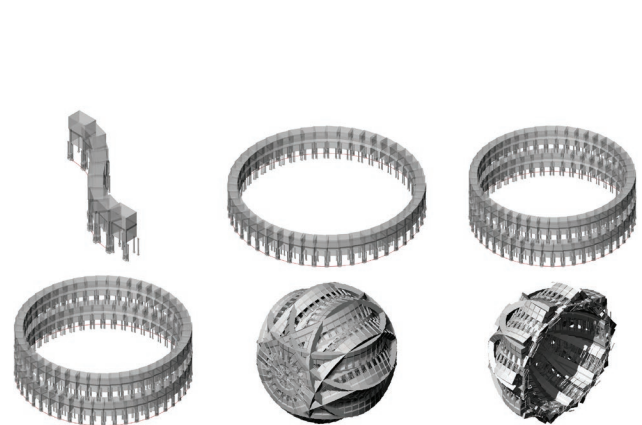


Figure 6. Selwyn Bacchus and Mitchell Willmarth, Palazzo dei Conservatori, Piazza del Campidoglio, Rome, Spring 2018. Phase 1 re-interpretation of the architectural order at the Conservatori using parametric software.

a historic building visible from their site then de-construct the architectural order as a code. For many historic buildings in Rome such as the church of Il Gesù the structural-technical and structural-symbolic are visibly the same (Figures 3-4).

In the second stage of Phase 1 students created an algorithm in Grasshopper, a parametric design tool widely used in the design industry. Students had to translate from the code used in the architectural order of the historic building to an analogous sequential code based on the programming procedures used in Grasshopper (Figure 5-6). Grasshopper allows designers an intuitive introduction to coding without requiring knowledge of programming or scripting (Figure 7). Grasshopper's visual programming environment was developed by David Rutten to be integrated with Rhino's 3-D modeling tools.³

METHODOLOGY: PHASE 2 – CITY (BUILDING) AS PHENOMENA

Concurrent with the code analysis and parametric re-coding we asked students to translate their observations of the site into a physical object. The focus on the phenomenal qualia of the site forced students to remain attuned to the physical-material presence of the site. One of the challenges working with parametric tools in design is to maintain or instantiate the link between the digital and physical as an immersive and experiential totality for what is finally a physical object: the building, urban space or environment created. Keeping students attuned to the physical but requiring them to objectify their observations instrumentalized their subjective perceptions (Figure 8). The objects were not replicas of the site or objects on the site, but representations of the shared perceptions/impressions of the team members. Developing the objects required discussion, analysis and the realization of the representation of their observations in a visual-physical form.

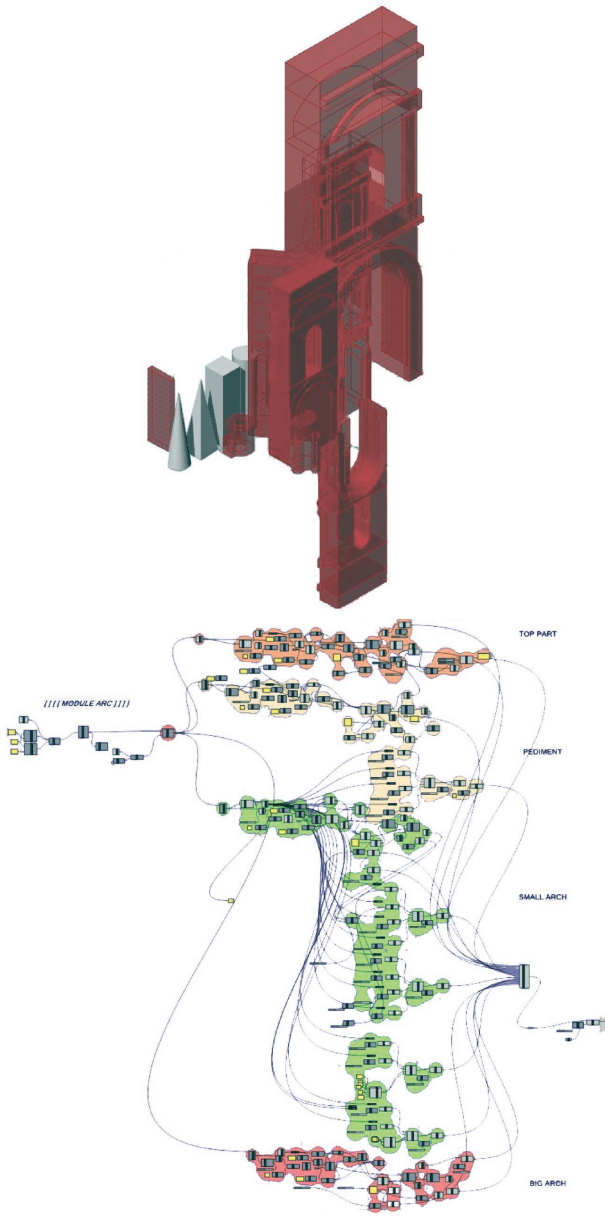


Figure 7. Colt Malloy and Robert Metz, Trajan's Market, Rome, Spring 2018. Phase 1 detail of semi-circular arcade Grasshopper code.

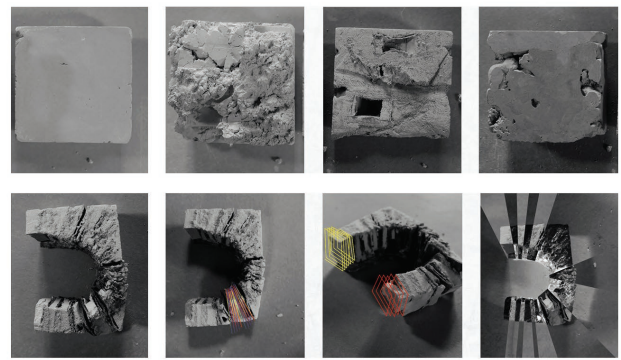


Figure 8. Tamara Stuart and Tamsan Mora, Ludus Magnus, Rome, Spring 2018, interpretive site photos montage and object interpretation of 'compression' as part of Phase 2 work.

METHODOLOGY: PHASE 3 – CITY AS SMART SYSTEM

Moving within the large framework of “smart city”, the goal of the third phase was offering students an opportunity to design a “smart building” considered as one component of the much larger system along the Via Papalis and in the city. Their proposed program was expected to work within an intelligent infrastructure able to provide services and solutions for Rome. The term ‘smart city’ is poorly defined, applied too broadly and lacks focus on specific technologies or sectors. What most smart city definitions have in common is the considered use of new technologies (usually ICT) and data as a means to solve the city’s economic, social and

environmental challenges. For us, the building program was considered part of a larger urban system encompassing smart city elements. The “performative” aspects of the building from a social, energy and environmental point of view, represented another key element of the design (Figure 9).

RESULTS AND CONCLUSIONS

One unexpected outcome of the studio approach was a significant increase in students’ awareness of historical buildings in Rome. Students correctly identified orders and their variations according to individual designers and patterns often

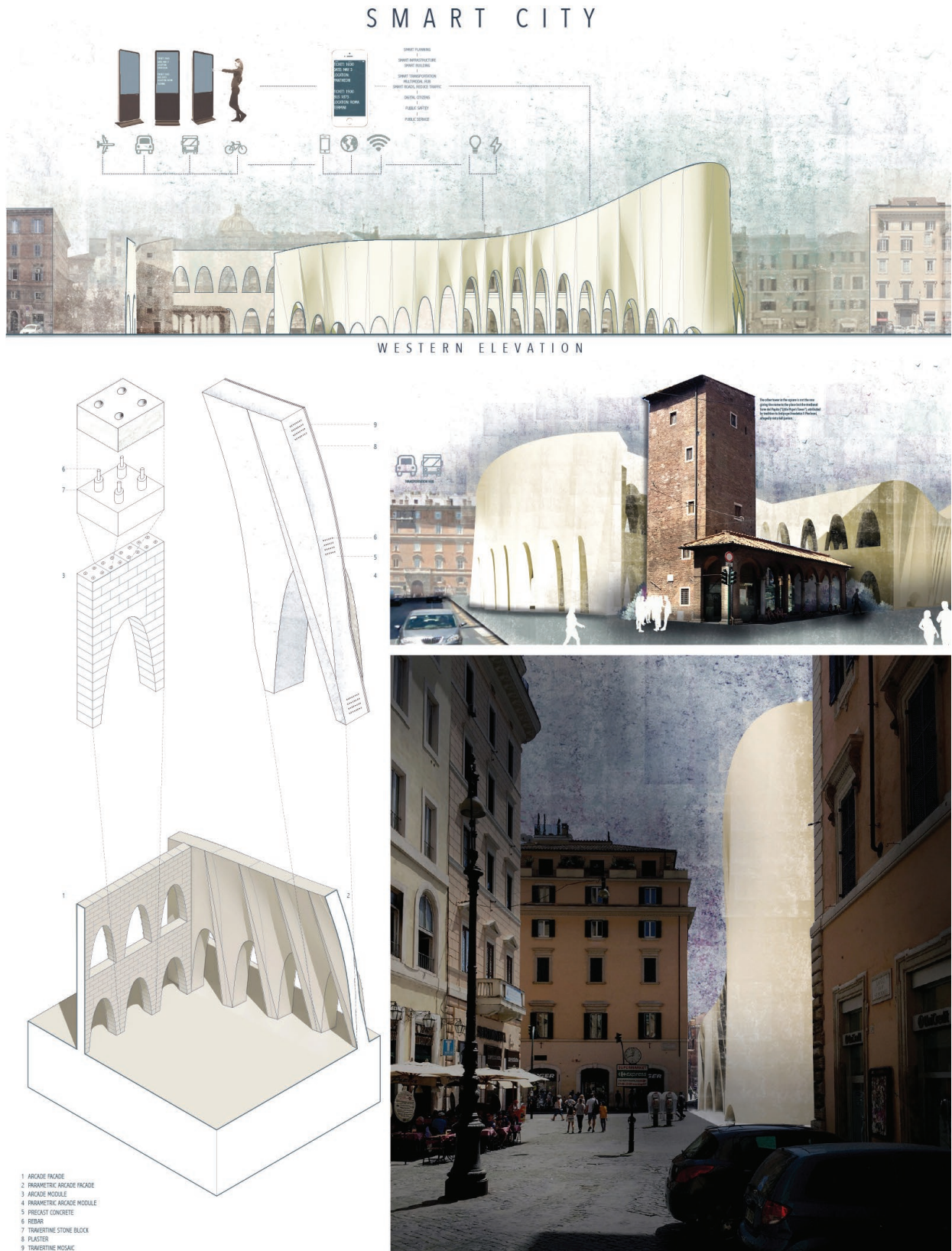


Figure 9. Riley Davis and Peter Slowik, Largo di Torre Argentina, Rome, Spring 2018. Multi-modal transportation center proposal for Largo Argentina developed as the program for the Smart City challenge on this site. The building form derives from an algorithm using transportation data and context analysis.

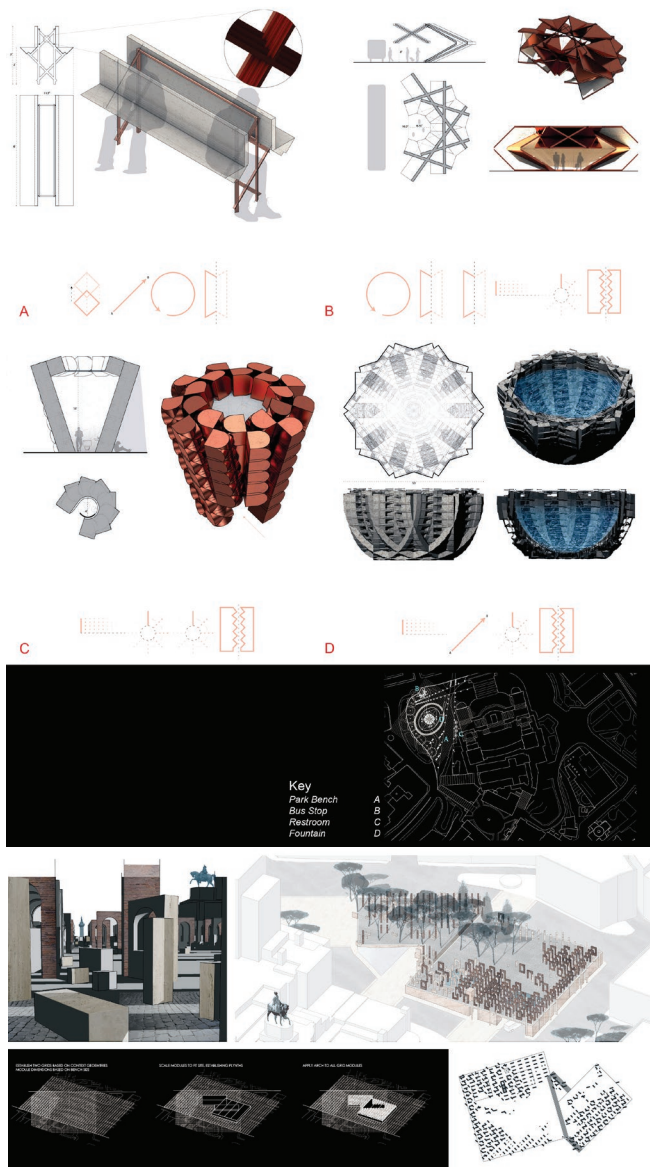


Figure 10. Selwyn Bacchus and Mitchell Willmarth, Piazza Venezia, Rome, Spring 2018. Phase 3 'Kit-bashing' using visual code developed from previous analysis of architectural orders creating urban furniture for proposed changes to Piazza Venezia. ode.

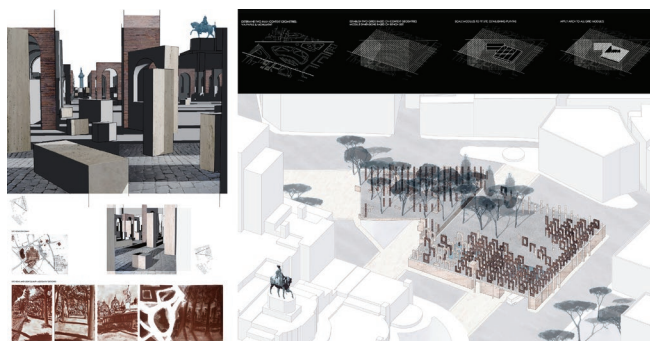


Figure 11. Rachel Filgas and David Sweere, Piazza Venezia, Rome, Spring 2018, Phase 3 procedural solution to create new piazza. Script based on a neutral sequence of physical and spatial procedures to generate formal organization of piazza components.

only apparent to historians. Future iterations of the studio would introduce culture as code to focus on the contingent and symbolic meaning of historical forms like the arch, or programmatic rituals associated with specific forms.

Final projects showed reasonable skill using the software, if not a fully developed understanding of algorithmic thinking as hoped. However, we recognized students began studio unfamiliar with the software and struggled with the fundamental principles of algorithmic thinking. There were successful projects both in terms of completion and ideation, just not always concurrent. The projects where students generated site-specific code in response to the design brief were most successful (Figures 10-11). The profession depends on digital technology throughout the design and fabrication process. Mastery of design thinking skills relative to algorithmic thinking is critical. Exposing students to the intellectual underpinning of coding and how it structures an approach to design thinking is decisive for professional architects. Finally, collaborative teaming was emphasized in the studio. Teams struggled a bit at the beginning, however; their overall success was encouraging and constant discussion about the value of collaboration as part of their professional education helped students understand the long-term value of the experience as part of the profession.

ENDNOTES

- 1 Grasshopper is a Rhino plugin. The introduction of Grasshopper was facilitated through a workshop given by Antonello Di Nunzio a local professional.
- 2 See, Heinrich Hübsch, ed., "The Principles of the Hellenic and Germanic Ways of Building with Regard to Their Application to Our Present Way of Building" in *In What Style Should We Build? The German Debate on Architectural Style* (Santa Monica, CA: Getty Center for the History of Art and the Humanities, 1992), 163.
- 3 Grasshopper is a graphical algorithm editor integrated with Rhino's 3-D modeling tools. Grasshopper does not require previous scripting knowledge, but the logic of the program uses the same algorithmic thinking applied to programming languages. See www.grasshopper3d.com.