KAIST Research Series

Ji-Hyun Lee *Editor*

A New Perspective of Cultural DNA

KAIST

Springer

KAIST Research Series

More information about this series at http://www.springer.com/series/11753

Ji-Hyun Lee
Editor

A New Perspective of Cultural DNA

Springer

Editor
Ji-Hyun Lee
Graduate School of Culture Technology
KAIST
Daejeon, Korea (Republic of)

ISSN 2214-2541 ISSN 2214-255X (electronic)
KAIST Research Series
ISBN 978-981-15-7706-2 ISBN 978-981-15-7707-9 (eBook)
https://doi.org/10.1007/978-981-15-7707-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Contents

About the Editor

Ji-Hyun Lee is an Associate Professor at the Graduate School of Culture Technology (GSCT) in Korea Advanced Institute of Science Technology (KAIST). She received her Ph.D. in School of Architecture (Computational Design) at Carnegie Mellon University writing a thesis about integrating housing design and case-based reasoning. Since joining the GSCT at KAIST, her research focus narrowed down to three interdisciplinary areas that are not mutually exclusive: (1) calculation for UX + service design, (2) cultural DNA with morphological analysis, and (3) computational creativity. These explorations result in computer-based frameworks or systems contributing to the enhancement of the calculability using algorithmic and/or heuristic computational methods. In other words, her research focus is on 'computational culture' as an extension of computational design.

She served for the Secretary of Computer-Aided Architectural Design Research in Asia (CAADRIA) from 2008 to 2010. Currently, she is the Director of the Information-Based Design (IBD) Research Group, Descartes Lab in KAIST and also serving as the Editorial Board Member for Architecture Research and International Journal of Innovations in Information Technology. She also serves as a Director of Korean Society of Service Design and Innovation (KSSDI), Korean Society of Design Science (KSDS) and HCI Korea. She is a member of Architectural Institute of Korea (AIK), Korea Intelligent Information System Society (KIISS), Society of CAD/ CAM Engineers and SIG-Design Creativity of the Design Society.

Simon's Ant: Towards New Task Environments for Design Alternatives

Robert Woodbury

1 The Task Environment Changes Behaviour

> *We watch an ant make his laborious way across a wind- and wave-molded beach. He moves ahead, angles to the right to ease his climb up a steep dunelet, detours around a pebble, stops for a moment to exchange information with a compatriot. Thus he makes his weaving, halting way back to his home....*
>
> *He has a general sense of where home lies, but he cannot foresee all the obstacles between. He must adapt his course repeatedly to the difficulties he encounters and often detour uncrossable barriers. His horizons are very close, so that he deals with each obstacle as he comes to it; he probes for ways around or over it, without much thought for future obstacles. It is easy to trap him into deep detours. Viewed as a geometric figure, the ant's path is irregular, complex, hard to describe. But its complexity is really a complexity in the surface of the beach, not a complexity in the ant....*
>
> *An ant, viewed as a behaving system, is quite simple. The apparent complexity of its behaviour over time is largely a reflection of the complexity of the environment in which it finds itself.*
>
> —Herbert Simon, *Sciences of the Artificial,* p. 63–64.

People are, of course, not ants. Cognitively, they are much more complex. As he argued, Simon's essential analogy still applies: observing people engaged in cognitive work yields, first, information about their task environment, and only second, information about their cognitive structure and abilities.

R. Woodbury (✉)
School of Interactive Arts and Technology, Simon Fraser University,
250-13450 – 102 Avenue, Surrey, BC V3T 0A3, Canada
e-mail: robw@sfu.ca

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_1

2 CAD Changes the Task Environment

An immediate implication for design is that design media, being a major part of the task environment, strongly impact designer action and behaviour. Yet the literature largely lacks accounts of such impact. Bhavani et al. (1999) argue that designers need specific training in the strategic use of design media in order to reify the capabilities inherent in such media. Flemming et al. (1997) argue that interaction metaphors inherited from past practice with manual drawing (in their case, the *T-square* metaphor) may hinder users from finding and using commands that have no analogue in manual techniques. In terms of task environment, the behaviour observed is partly inherited from prior media and partly influenced by the digital media commands and structure. Bilda and Demirkan (2003) explain the differences between manual and digital media in terms of media differences (CAD does not provide doodling, diagramming and pencil gesture), between the relative state of development of the media (sketching is mature, CAD operations are primitive), and in subject relative experience with the two media forms (all subjects were nearly novice CAD users). There is thus little theory that helps predict how change to design media results in change to design. Further, for a central part of design work, that is, exploring for alternatives ("search in a problem space" in Simon's terms), both exemplary systems and guiding theory are in particularly short supply.

We are concerned here with designers using alternatives in their work, particularly with creating new tools that help designers create and understand alternatives. If Simon's ant analogy holds, we should be able to observe different patterns in using alternatives across design media. The most widely accepted general pattern for expert designers using alternatives describes a general-to-concrete hierarchy of design problems with a predominant breadth-first exploration at each level of the hierarchy, followed by a depth-first exploration of one or more chosen alternatives. This breadth then depth pattern may be repeated at any hierarchical level. This process goes by several names, for example, Akin's (2001) *depth-and-breadth-first search* and Fricke's (1996) *stepwise design strategy* and *balanced search tactic*. Few studies, notably Fricke (1996), though, have been conducted over time frames sufficient to reveal this pattern being played out through a design process. For example, Akin (1986) describes a study of a short time scale suitable for protocol analysis. Smith and Tjandra (1998) describe exercises of "a few hours". Goldschmidt (1991) studied episodes of "one to two hours". Further, the design media used in most studies are manual.

3 Working with Alternatives

It is beyond argument that designers work with alternatives.

> *Every idea that is a true idea has a form, and is capable of many forms. The variety of forms of which it is capable determines the value of the idea.*
> —Frank Lloyd Wright

As Wright implies, exploring a space of possibilities is central, indeed essential, to many kinds of complex work. Accounts of such exploration have occurred in the literature for a long time, with perhaps the first thorough systematic treatment being that of Newell and Simon's Human Information Processing System (HIPS) (Newell and Simon 1972). HIPS describes human problem-solving action as being search in a problem space, largely constrained by a task environment. In turn, the task environment almost always includes external media with which people store problem configurations. In design, HIPS became a principal research concept, around which arose accounts of designer action (Akin 1986; Cross 2004, 2001, 2008), the use of heuristic search as a computational strategy for solving design problems (Eastman 1973; Pearl 1984; Heisserman 1994; Flemming et al. 1992), and direct use as a concept and object in systems that aim to support design work (Chandrasekaran 1990; Woodbury and Burrow 2006). The idea of a space of designs figures large in the significant literature on shape and spatial grammars. In visual analytics, the term *analysis of competing hypotheses* describes a key phase in analytic workflows (Heuer 1999). The first step in such an analysis is to identify all potential hypotheses, though this is one of the least specific aspects of the method. Relatively recently, the computer science community, and HCI in particular, has published a number of perspectives, system and evaluations. For example, Shneiderman et al.'s (2000, 2006) argument for supporting exploration and providing rich history-keeping appears to have provided key direction in this area. Each field has its own terminology and authors such as Lunzer and Hornbæk Lunzer and Hornb (2008) have coined now-accepted terms such as *subjunctive interfaces*. In the face of a Tower-of-Babel-like profusion of terms, we adopt the simple and, hopefully, neutral term *alternatives* to encompass the entire complex. When we write "alternatives" we refer to the objects representing designs in the general enterprise of exploring a space of possibilities.

While there is a significant literature confirming the usefulness of supporting alternatives in computer-aided design systems, there is yet but a small set of distinct ideas on how to do so. In systems used on a regular basis, support for alternatives may exist, but is always a secondary feature, typically aimed at making a small number of variations of a design. What has been called the *Single State Document Model* (Terry and Mynatt 2002) dominates CAD interface designs.

> *The Single State Document Model requires a document to be in one, and only one, state at any particular time, thereby imposing a serial, linear progression through a task that is at odds with the "messy", highly iterative creative process.*
> —Terry and Mynatt (2002).

Thus, how people work (designing with and through alternatives) clashes with the media they use (based on the Single State Document Model). It follows that discovering and devising ways to support alternatives is an appropriate goal.

4 What is an Alternative?

But what is an alternative? Is the idea of an alternative constant across tools?

The diversity of the literature on supporting alternatives suggests that constructs to recognize or measure an alternative vary across design media. Theory concurs—a given task environment makes some things easy and others hard, and we can expect to see people employ the "easy" more frequently than the "hard". As a practical matter, let us consider only those design alternatives that appear in an external medium—leaving out those inaccessible and empirically doubtful ones that are "in the head". Then, an alternative is a "mark" on a physical medium or a symbol structure in a digital medium. Consider a few examples.

Figure 1 shows a sketch study "*Study for the Trivulzio Equestrian Monument,*" made by Leonardo da Vinci in 1508–10, with a total of four different revisions of the sketch, all drawn by the artist on a single sheet of paper (da Vinci 1508). Here, an alternative is, arguably, a composite of several sketch fragments, and the entire sketch "contains" or connotes multiple alternatives. It is likely more plausible to interpret the contents of the multiple sketch fragments less as specific proposals for the positions of heads and legs as more as suggestions for exploration as described by Buxton (2007, pp. 111–120). Nonetheless, we see here sets of superimposed sketch fragments juxtaposed with other such sets within a single overall composition. Clearly, this represents a distinct and media-dependent technique for representing alternatives.

Figure 2 shows, in a single sheet, a so-called "ideation study" for running shoes, with multiple options. Such pages can be produced in a very short time by skilled designers. In this example, alternatives are spatially juxtaposed, likely for the purpose of suggesting comparisons and recombination of features into new versions. Interspersed with overall sketches are details that elaborate in particular features of one or more alternatives. Though seldom reproduced in publications, designers also often use multiple layers of transparent paper to superimpose alternatives.

Cross (2011, page 17) and Lloyd and Snelders (2003) show a series of hand-drawn sketches on a single sheet of paper by Philippe Starck, produced in designing the "Juicy Salif" citrus squeezer design. The designer arrived at the final product design by working his way through a sequence of about thirty related sketches and ideas (the lemon, the squid, and the 60's rocket/spaceship). Our main reason to include this example is that it can be interpreted as successive refinements of a single idea that was present from the outset. One idea emerged later: the spaceship. The sketch can thus be taken as a counter-argument against alternatives as is merely demonstrates design refinement. We offer two observations to this counter-argument. First, Starck devised the final design while being able to view and compare several alternatives

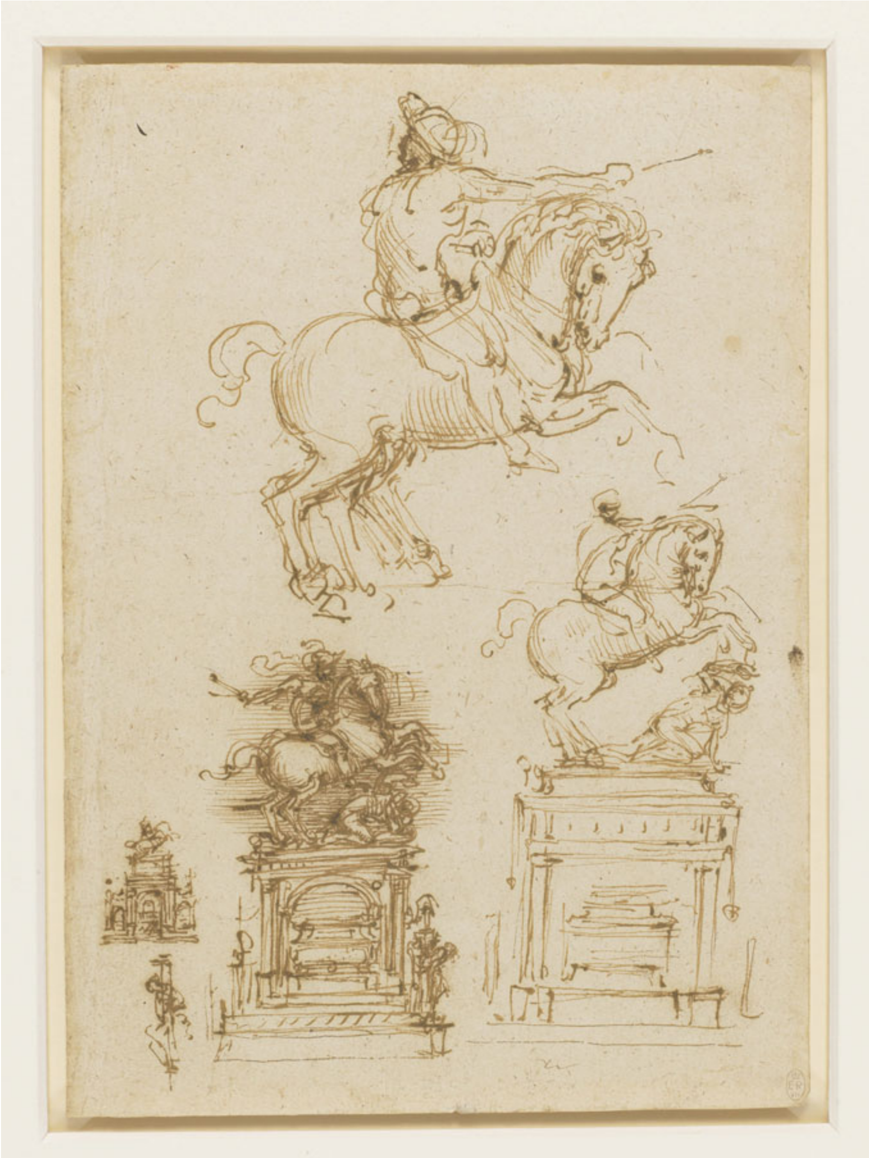


Fig. 1 Examples of alternatives "in the wild" in the field of visual arts: "*Study for the Trivulzio Equestrian Monument*" (da Vinci 1508) by Leonardo da Vinci. Different parts of this sketch feature multiple alternative outlines, with different positions and orientations (see the cutouts on the left side of the figure, from top to bottom): (1) different rider's head positions, (2) multiple positions of the horse's hind legs, and (3) multiple positions of the horse's front legs

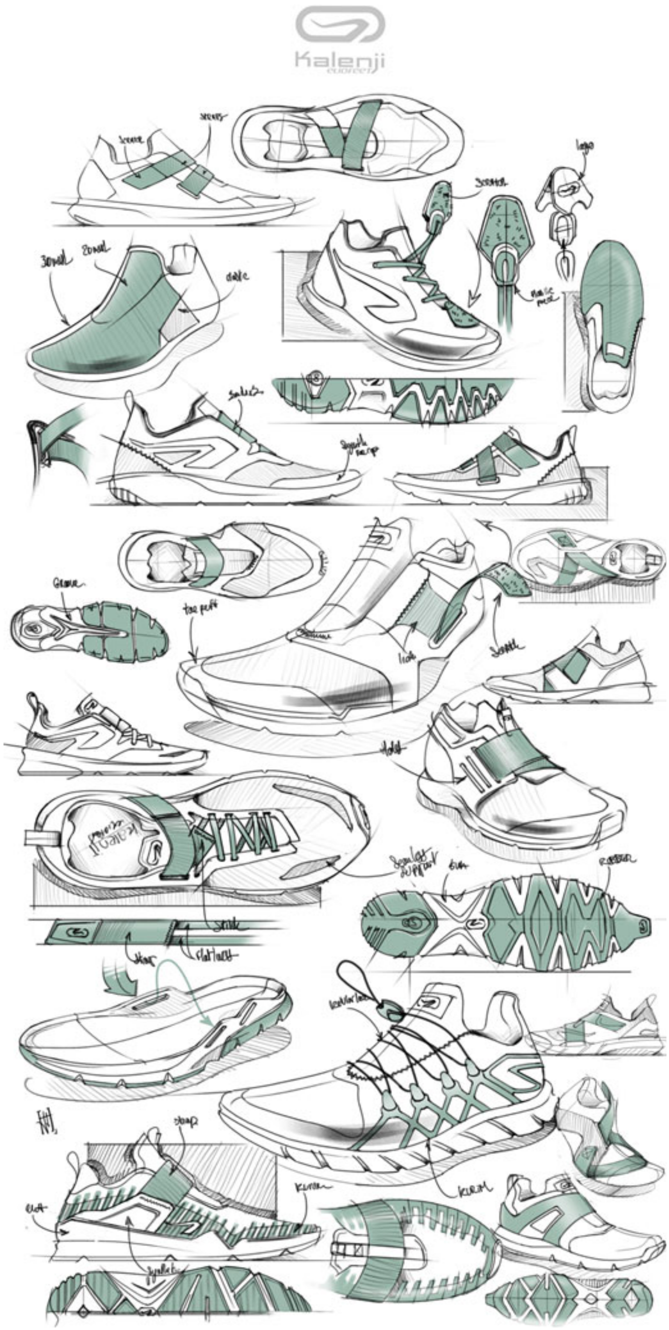


Fig. 2 A page of sketches exploring concepts for running shoes. Marc van Tichelen, designer. Reproduced with permission

at once, and then recombine elements of several promising designs into the final (and commercially successful) product design. Juxtaposition is relevant even in a process of successive refinement. Second, design refinement is a limit form of using alternatives. That decisions are progressively made does not obviate the use of (or need for) multiple alternatives in the supporting media. (Remember, we use the term "alternatives" in a deliberately inclusive sense to refer to distinct representations used in design.)

Digital media, not surprisingly, provides different examples of alternatives. Designers using systems bound by the Single State Document Model employ well-known adaptations such as using multiple versions of files, storing partial alternatives on separate layers or cutting and pasting entire alternatives into an overall spatial scene. When using representations that explicitly model multiple states, the differences become both richer and more nuanced. We include two archetypal examples here, from generative and parametric design, respectively.

The label "generative design" applies to a broad range of ideas and techniques, from which we highlight spatial grammar (we include both shape grammar and the wider class of grammars defined over spatial representations other than shapes).

Spatial grammars require a *rule set* and a *starting shape,* and produce a *language of designs* by recursively applying rules beginning from the starting point. Chains of such rule applications are called *derivations* or *derivation sequences.* Spatial grammars thus produce a space of representations, linked into derivations through rule applications. Both the language of designs and all interim productions (called *sentential forms*) are candidates for being considered as alternatives. All of these devices: rules, applications, derivations and (subsets) of the grammar language may be used to explain a particular shape grammar (as shown in (Koning and Eizenberg 1981, 2019). For instance, Figure 3 shows members of the language of designs implied by John Portman's own house.

Parametric models produce variations through changing model inputs. Figure 4 shows that models so produced can be visually diverse, even though produced by "mere" parametric change. Arrays such as that shown in Figure 4 have a regular part of architectural presentations and publications in recent years.

Both of these digital examples share the property that alternatives are somehow marked or signified by designer intention (as they must be in the manual examples because the designer made the effort to commit each alternative to "paper)". In spatial grammar, it is typical to call out specific members of the language of a grammar as somehow representative or typical of the whole. It is quite uncommon to have a meaningful grammar with only a few members of its language. Parametric models can produce very large numbers of variations, not each of which is a meaningful alternative. Though the representation provides for these large numbers, interfaces mostly remain in the Single State Document Model. Designers are typically reduced to using ad-hoc devices for alternatives similar to those noted above.

We conclude that an alternative is something produced in a design representation and about which a designer cares enough to mark in some way. This latter point will certainly be critical in interface design.

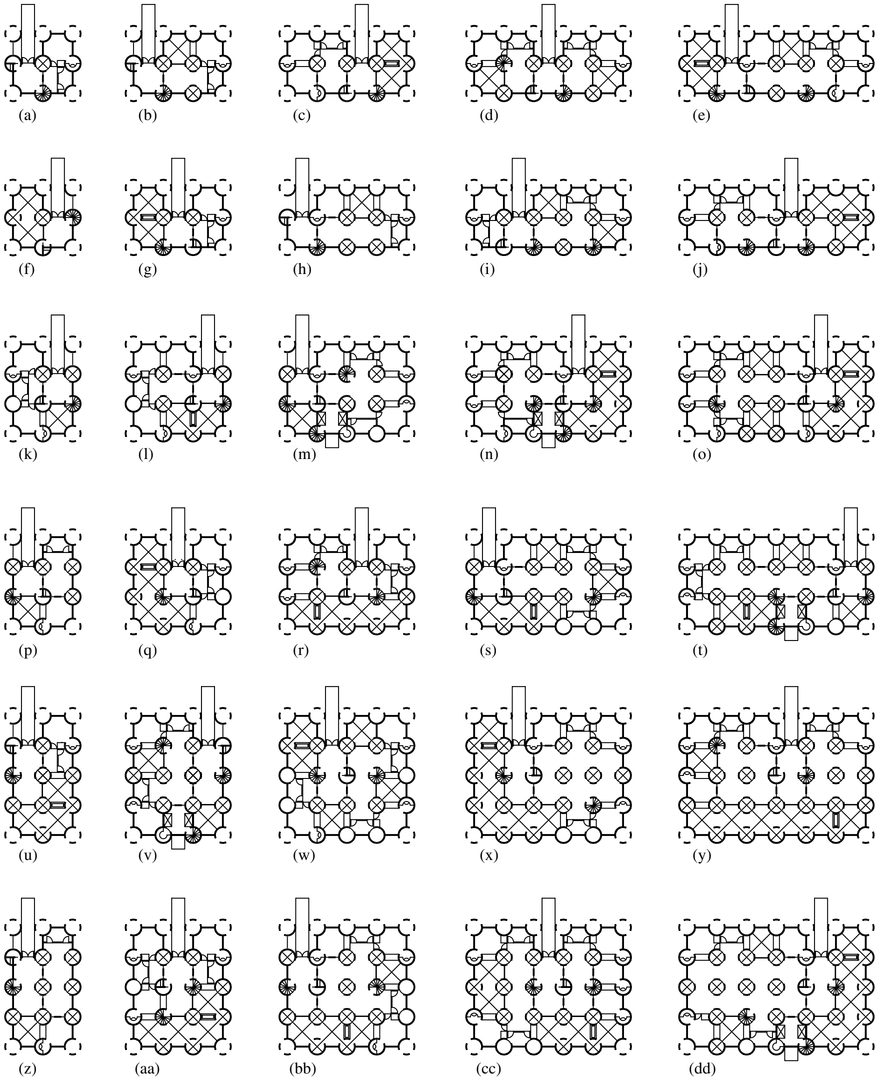


Fig. 3 Selected members from the language of designs of a grammar based on John Portman's own house. Figure reproduced with permission from Ligler and Economou (2019)

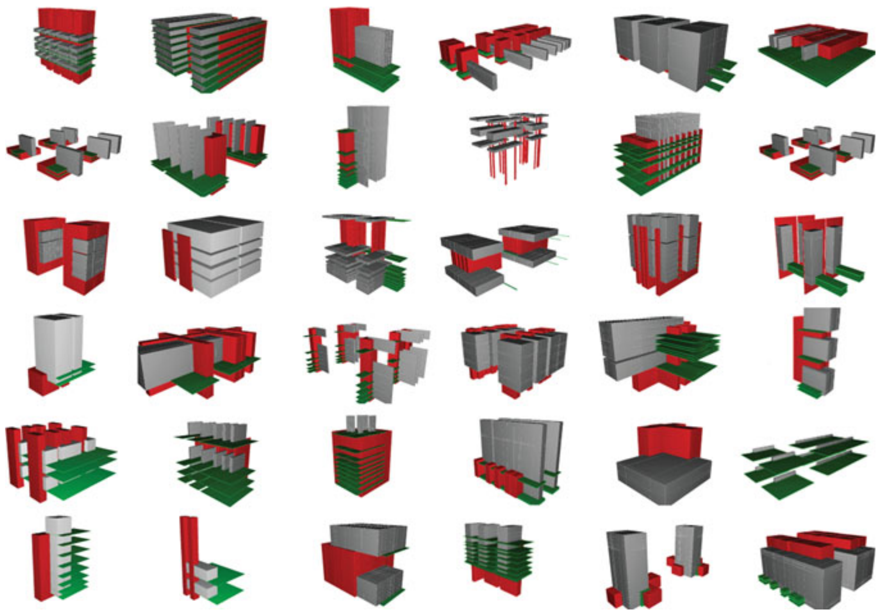


Fig. 4 Array of parametric alternatives produced from the same model. Used with permission from Shireen et al. (2017)

5 Supporting Design Alternatives

Each tool provides its own means to record alternatives, for example, sketches employ superposition, copy and modify, rapid sketch creation; conventional CAD provides aggregation and manipulation commands; and parametric CAD provides model editing and parametric variation.

Once a tool is mastered, we should expect designers to do different things with each tool. Indeed, we should be surprised if actions are comparable across tools. For instance, there is no easy analogue to a parametric variation in a sketch, nor does parametric modelling support the rapid, plentiful ambiguity of sketches in Buxton's sense (2007). It may even be that comparable actions across tools are the result of tool designers pursuing a skeuomorphic design strategy, for instance, the reproduction of sketching affordances that was a goal in many 1980s to 1990s CAD papers and even more recently in computer graphics Bae et al. (2009). Nor should we expect the notion of an alternative to be invariant across tools. For example, parametric variations may seem trivial as alternatives, but such variations of a model can yield dramatically different outcomes.

This expectation of difference flows into research. Given different tool affordances and capabilities, creating good constructs for the concept of a design alternative may prove elusive. While the simple act of moving a slider may generate hundreds of parametric variations in a few seconds, it may seem odd to put these on the same conceptual footing as a separately constructed pencil sketch of a design alternative. Or is it? Experts in parametric tools regularly employ a *deferment strategy* (Woodbury 2010, p. 43) to construct models that allow them to explore alternatives later. Thus, seemingly simple variations may be part of an explicit exploration strategy. We can expect that the signal for a design alternative will vary widely across design media.

Thus, researchers and tool developers in design alternatives face a dilemma. New and old tools do widely (and hopefully, wildly) different things. Returning to Simon's ant, different tools form different task environments and thus may (and should, if tool developers succeed) change designer behaviour radically from tool to tool. One horn of the dilemma attempts to compare tool use across manual and digital media, for example, (Bilda and Demirkan 2003), arriving at conclusions that the media vary, but raising serious internal validity issues. The other horn of the dilemma abandons comparison between old and new, making it difficult to argue that progress is being made.

These deep issues of research validation aside, what are potentially useful system features for supporting design alternatives?

General frameworks and descriptive accounts give a basis for understanding alternatives in design, but our interest lies in empowering designers with new kinds of tools. Research on this issue proceeds by cases: exemplary systems, their evaluations and analyses based upon them. And the action is in the domains: a system has to do something specific, for instance, programming, data analysis or building design. This focus on domains may explain why there is no single literature on alternatives—publications are spread over disciplines with limited cross references. A major problem with domains is that they tend to invent *lenses,* that is, particular interface designs that persist across multiple cases. These lenses focus effort on a few interface ideas and features, largely excluding others from consideration. Thus, the alternatives literature largely reports on a few basic designs. As I have argued (Woodbury 2016), lenses strongly channel system designs to consider only limited aspects of the entire problem. Thus, one should largely look for ideas that transcend or abstract the lens from which they come. I have identified six principal lenses into which almost every reported system falls: grammar, history, version, representation, task and search. For example, the grammar lens takes the intellectual structure of a generative grammar as a machine over which to build an interface. It uses ideas such as rule, derivation and derivation path to structure the interface. I use lenses largely as a filter, to look for system features that either (or both) depart from the lens or occur across several lenses. I take these features as candidates for further design and development. At a primitive level, they include juxtaposition (putting alternatives side-by-side), superposition (combining alternatives into single views), rapid serial juxtaposition (commonly used in choice interfaces in digital games), abstracting into charts and graphs (from which can be built more complex tools such as small multiples and multi-dimensional Pareto charts), and semantic zoom (from glyph to full CAD model). Larger, composite features emerge. Parallel editing enables single editing actions to affect multiple alternatives (Zaman et al. 2015). Exploring a local search space supports reuse of past design decisions (Lunzer 1994; Woodbury et al. 2000; Woodbury 2010; Zaman et al. 2015). User-defined and -controlled collections of alternatives are a chief feature of our recent prototypes (Sanchez et al. 2012; Kolarić et al. 2014; Zaman et al. 2015; Woodbury et al. 2017; Kolarić et al. 2017; Mohiuddin et al. 2017). Recent work (Shireen et al. 2017) shows that users need and invent such collections when tasked with understanding and organizing large numbers of alternatives.

All of these ideas stand apart from the most frequent approach in the literature for using digital alternatives in design. Papers taking this stance employ a heuristic search algorithm to sample a typically informally specified design space and argue that the outputs of such a process are useful in design. There are many such papers, using, for example, evolutionary algorithms, simulated annealing, Pareto optimization, tabu search, dispersion sampling and (more recently) generative adversarial networks. I label research of this type as *appealing to an oracle.* In computer science, an oracle machine is a Turing machine augmented by a black box, an *oracle,* able to solve a decision problem in a single operation. In mythology, an *oracle* can divine the future, typically by appeal to a higher power. To appeal to an oracle is to accept what the oracle produces as useful and not to inquire about how it does its work. From a research perspective, oracles are attractive as low-hanging fruit: undertaking such a project follows an established path, can be done with modest effort and yields a demonstrable result. As research contributions though, two problems emerge. The first is that such works are incremental: they elaborate one particular approach to using alternatives in design, whereas the domain may need new approaches. The second problem is that the approach itself is flawed: it does not support design tasks as they are done in the wild.

Why not? I offer two arguments. The first is that designers design—they devise things to achieve goals. In the task of designing, a design's role is to be critiqued and re-worked in response. Schön's (1983) account of the reflective practitioner argues this invariant thoroughly. Thus, the almost inevitable fate of an oracular result is to be subjected to yet more design work. Bradner et al. state this well in their study of designers using optimization systems.

> *Professionals reported that the computed optimum was often used as the starting point for design exploration, not the end product.*
> —Bradner et al. (2014).

The second argument is that designers explain their work. In such explanations agency is important. "This is what we thought and did" presents a more credible argument than "This is what the algorithm told us". Although facetiously expressed, the way designers explain and understand (explain to themselves) designs involves having worked with and through the design's ideas over time. Both of these arguments imply a strong need for basic and practical knowledge of designing interactions and systems supporting design alternatives.

6 How Malleable Are Task Environments?

Clearly, changes in tools can have variable effects on task environments. A refinement of an existing tool is less likely to induce a major change in designer action than did, for instance, the introduction of data flow visual programming in parametric design. Are there limits to changing the task environment? For instance, if appealing to an oracle can produce appropriate designs, then designers will no longer search in a problem space; rather, they will consult an oracle and select from the results. It would appear that tools may be able to change the fundamental nature of human problem solving and thus design. I would suggest that there are limits to such change, and further, that there is a hierarchy of task environment features that is hard to change at the bottom and more easy at the top. In other words, there are both variants and invariants in the situation. But what are these? One source for such is the design situation itself, particularly the size of design space, the structure of design problems and the role of knowledge.

Design spaces are VAST (in Dennett's terms Dennett (1995)), that is, they will defeat any attempt at definitive enumeration. As a consequence, designers (and oracles) satisfice rather than optimize Simon (1956). A likely invariant that arises is a need to record, organize and visualize multiple potentially satisficing solutions as they arise. Adding to this need for multiplicity is that designs are seldom judged against a single criterion; thus, a design medium should be able to compare alternatives with multiple criteria. Likely a major invariant is that satisficing and multiple criteria act in concert to demand that oracles be wrapped in direct interaction. Since designers seldom take oracular results as final, whatever an oracle produces must be subject to the same interaction as any other design.

The word "design" labels many processes and thus structures across many domains. At the risk of implying a false spectrum, a design problem may be sufficiently well specified that hill-climbing strategies against fixed goals will reach satisficing solutions. Another design problem may be ill-structured or wicked; it may defy full or precise definition as the problem varies depending on the solution proposed. Indeed, there exists a significant literature that characterizes design problems by type; it suffices here to note that design processes and thus useful tools differ across problems and design domains. We should not expect a universal nor small set of operators for creating and visualizing design alternatives.

Designers use several forms of knowledge varying from formal or articulate (able to be explicitly specified) to tacit (incompletely codified, often embodied, known through experience). While formal knowledge can be directly codified in a design medium (for example, output parameters in a parametric model), tacit knowledge must be recognized. A task environment for alternatives must thus support multiple coordinated views, some expressing formal design knowledge and others providing general views through which tacit concepts, patterns and features can be seen.

The design medium itself provides variants in task environments. For example, manual sketching, image processing systems and parametric models are all useful design media, but each suggests and supports very different operations and visualizations. We should not expect that what works in one medium to transfer much to any other. Rather those who seek to support alternatives in a particular design medium would do well to immerse themselves in what designers actually do with that medium, singly and in concert with other media.

In summary, we know designers work with alternatives for which current systems provide impoverished support. We should expect such support to be specific to the design media (e.g., parametric models) being supported, and to be used by designers in ways both specific to the media and unexpected. While several system features seem promising and some have been reasonably well tested, the domain is young. There is much still to be learned about interactive design space exploration.

Acknowledgements This work was supported by the Natural Sciences and Engineering Research Council of Canada, Bentley Systems Inc., and Smartgeometry.

Conflict of Interest The author declares that he has no conflict of interest.

References

Akin, Ö.(1986). *The psychology of architectural design*. London: Pion.
Akin, Ö. (2001). Variants in design cognition. In C. M. Eastman, W. M. McCracken & W. C. Newstetter (Eds.), *Design knowing and learning: Cognition in design education* (pp. 105–124). Elsevier Science, Oxford.
Bae, S. H., Balakrishnan, R., & Singh, K. (2009). EverybodyLovesSketch: 3d sketching for a broader audience. In: *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (pp. 59–68). New York, NY, USA: ACM. UIST '09.
Bhavnani, S. K., John, B. E., & Flemming, U. (1999). The strategic use of cad: An empirically inspired, theory-based course. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp 183–190). ACM
Bilda, Z., & Demirkan, H. (2003). An insight on designers' sketching activities in traditional versus digital media. *Design Studies*, *24*(1), 27–50.
Bradner, E., Iorio, F., & Davis, M. (2014). Parameters tell the design story: Ideation and abstraction in design optimization. In R. Goldstein &, Gerber D. D. (Ed.), *Symposium on Simulation for Architecture and Urban Design: The Society for Modeling and Simulation International* (pp. 77–84).
Buxton, B. (2007). *Sketching user experiences: Getting the design right and the right design*. Morgan & Kaufmann
Chandrasekaran, B. (1990). Design problem solving: A task analysis. *AI magazine*, *11*(4), 59.
Cross, N. (2001). Design cognition: Results from protocol and other empirical studies of design activity. In *Design knowing and learning: Cognition in design education* (pp 79–103). Elsevier
Cross, N. (2008). *Engineering design methods: Strategies for product design*. Wiley.
Cross, N. (2011). *Design thinking: Understanding how designers think and work*. Berg.
Cross, N. (2004). Expertise in design: An overview. *Design Studies*, *25*(5), 427–441.
da Vinci, L. (c. 1508-10). Study for the trivulzio equestrian monument. https://www.rct.uk/collection/912355/studies-for-the-trivulzio-monument. Accessed on 17 January 2020.
Dennett, D. C. (1995). *Darwin's dangerous idea: Evolution and the meanings of life*. Simon & Schuster.
Eastman, C. M. (1973). Automated space planning. *Artificial Intelligence*, *4*(1), 41–64.

Flemming, U., Baykan, C., Coyne, R., & Fox, M. (1992). Hierarchical generate-and-test vs. constraint directed search: A comparison in the context of layout synthesis. In J. Gero & F. Sudweeks (Eds.), *Artificial Intelligence in Design '92* (pp. 817–838) Dordrecht, the Netherlands: Kluwer Academic Publishers.
Flemming, U., Bhavnani, S. K., & John, B. E. (1997). Mismatched metaphor: User vs system moeel in computer-aided drafting. *Design Studies*, *18*(4), 349–368.
Fricke, G. (1996). Successful individual approaches in engineering design. *Research in Engineering Design*, *8*(3), 151–165.
Goldschmidt, G. (1991). The dialectics of sketching. *Creativity Research Journal*, *4*(2), 123–143.
Heisserman, J. (1994). Generative geometric design. *IEEE Computer Graphics and Applications*, *14*(2), 37–45.
Heuer, R. J. (1999). *Psychology of intelligence analysis*. Center for the Study of Intelligence.
Kolarić, S., Erhan, H., & Woodbury, R. (2014). CAMBRIA: A tool for managing multiple design alternatives. In *Proceedings of the 2014 Companion Publication on Designing Interactive Systems* (pp. 81–84). New York, NY, USA: ACM.
Kolarić, S., Erhan, H., & Woodbury, R. (2017). CAMBRIA: Interacting with multiple CAD alternatives. *Computer-Aided Architectural Design* (pp. 81–99). Future Trajectories: Springer.
Koning, H., & Eizenberg, J. (1981). The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B*, *8*(3), 295–323.
Ligler, H., & Economou, A. (2019) From drawing shapes to scripting shapes: Architectural theory mediated by shape machine. In *Simulation for Architecture and Urban Design* (pp. 279–286).
Lloyd, P., & Snelders, D. (2003). What was Philippe Starck thinking of? *Design Studies*, *24*(3), 237–253.
Lunzer, A. (1994). Reconnaissance support for juggling multiple processing options. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (pp. 27–28). ACM.
Lunzer, A., & Hornbæk, K. (2008). Subjunctive interfaces: Extending applications to support parallel setup, viewing and control of alternative scenarios. *ACM Transactions on Computer-Human Interaction (TOCHI), 14*(4), 17:1–17:44
Mohiuddin, A., Woodbury, R., Cichy, M., Mueller, V., & Ashtari, N. (2017). A design gallery system: Prototype and evaluation. *ACADIA 2017: Disciplines & Disruption* (pp. 414–425). Boston, MA: ACADIA.
Newell, A., & Simon, H. (1972). *Human Problem Solving*. Englewood Cliffs, N.J.: Prentice-Hall Inc.
Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving*. Inc, Reading, MA: Addison-Wesley Pub. Co.
Sanchez, R., Erhan, H., Woodbury, R., Mueller, V., & Smith, M. (2012). A visual narrative of parametric design history. Aha! Now I see how you did it! In *Proceedings of the 30th eCAADe Conference* (pp. 259–268).
Schön, D. (1983). *The reflective practitioner: How professionals think in action*. Basic Books.
Shireen, N., Erhan, H., Woodbury, R., & Wang, I. (2017). Making sense of design space: What designers do with large numbers of alternatives? *Computer-Aided Architectural Design* (pp. 191–211). Future Trajectories: Springer.
Shneiderman, B. (2000). Creating creativity: User interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *7*(1), 114–138.
Shneiderman, B., Fischer, G., Czerwinski, M., Resnick, M., Myers, B., Candy, L., et al. (2006). Creativity support tools: Report from a US National Science Foundation sponsored workshop. *International Journal of Human-Computer Interaction*, *20*(2), 61–77.
Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review*, *63*(2), 129.
Smith, R. P., & Tjandra, P. (1998). Experimental observation of iteration in engineering design. *Research in Engineering Design*, *10*(2), 107–117.

Terry, M., & Mynatt, E. D. (2002). Recognizing creative needs in user interface design. In *Proceedings of the 4th Conference on Creativity and Cognition* (pp. 38–44). New York, NY, USA: ACM. C&C '02, ACM ID: 581718

Woodbury, R. (2010). *Elements of parametric design*. Taylor and Francis, with contributions from Brady Peters, Onur Yüce Gün and Mehdi Sheikholeslami.

Woodbury, R. (2016). The grammar lens: How spatial grammar channels interface design. In J. H. Lee (Ed.), *Morphological analysis of cultural DNA* (pp. 199–226). Singapore: Springer.

Woodbury, R. F., & Burrow, A. L. (2006). Whither design space? *AIEDAM, Special Issue on Design Spaces: The Explicit Representation of Spaces of Alternatives*, *20*, 63–82.

Woodbury, R., Datta, S., & Burrow, A. (2000). Erasure in design space exploration. *Artificial Intelligence in Design*, *2000*, 521–544.

Woodbury, R., Mohiuddin, A., Cichy, M., & Mueller, V. (2017). Interactive design galleries: A general approach to interacting with design alternatives. *Design Studies*, *52*, 40–72.

Zaman, L., Stuerzlinger, W., Neugebauer, C., Woodbury, R., Elkhaldi, M., Shireen, N., & Terry, M. (2015). GEM-NI: A system for creating and managing alternatives in generative design. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 1201–1210). New York, NY, USA: ACM.

Trust Considerations in the Coordination of Computational Design Teams

Alparslan Emrah Bayrak

1 Introduction

As systems increase in complexity, the amount and importance of coordination effort to make effective design decisions for these systems also increase. Current enterprises that develop complex systems (Burton et al. 2011) or engineering design approaches such as multidisciplinary optimization (Martins and Lambe 2013) are built around the idea of partitioning the overall problem into manageable subproblems that can be solved by different people or engineering methods. This partitioning approach creates interdependent (or coupled) subproblems whose solutions depend on each other. These interdependencies require a coordination effort between decision-makers (human or computer agents) who are responsible for solving these subproblems to produce a coherent whole that can achieve the desired system-level characteristics (National Aeronautics and Space Administration 2008).

Trust is an essential factor for effective team coordination. The aforementioned couplings between subproblems necessitate the team members to rely on the decisions made by others. Trust determines whether those couplings are reflected accurately in the decisions of the team members. In an extreme case where there is no trust between two team members who work on two coupled problems, the decisions will be made as if these problems are uncoupled since neither of the team members will be willing to take into account the other one's decisions. The negative implications of distrust in teams have been highlighted in several studies (Jong et al. 2016). Low levels of trust in a team overall may lead teammates to ignore essential information coming from other members, increase the time to reach agreements in teams and hence decrease the team performance.

A. E. Bayrak (✉)
School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030, USA
e-mail: ebayrak@stevens.edu

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_2

On the other hand, a healthy amount of distrust might be beneficial for the team performance especially when the quality of the decisions made by others are questionable. The quality of design decisions is highly affected by the accuracy of the experimental data or computational models that they are based on. However, these information sources may have several types of uncertainties that stem from measurement errors in experiments or a lack of knowledge of the system. Such uncertainties lead to unreliable decisions trusting in which may lead to suboptimal solutions for the overall system.

Here, we discuss a notional case study to quantitatively analyze the impact of trust on overall team performance during problem coordination. While trust is a human belief, we look at this problem from a purely computational perspective where a team consists of a set of computational agents each of which solves an individual optimization problem that is coupled with the problems of other agents. The agents make design decisions under uncertainty in their information source. We model trust as the confidence of an agent to any other agent's decisions and model the team coordination as an optimization problem where one of the agents is responsible for the coordination task using the existing decomposition-based design optimization approaches. Using this framework, we demonstrate how much the "coordination agent" should trust in the "design agents" for optimal team performance on a notional example, rather than modeling how trust naturally emerges and evolves in a real scenario.

2 Related Work

This section presents a brief review of the relevant literature on problem coordination from design optimization field and the approaches to model trust in teamwork. This brief review is partial and only intended to highlight a gap in the respective literatures.

2.1 Analysis of Trust in Teams

Trust is a human belief that plays an important role in group interactions and the effectiveness of teamwork. Several definitions of trust exist in the literature (Ramchurn et al. 2004). The studies in the literature discuss human trust generally in three dimensions: benevolence, honesty, and competence (Mayer et al. 1995). While in human to human trust, all three are important, in human to computer relationships, competence becomes the dominant factor determining how trust is formed. In the present study, we focus on this third dimension and simply refer to trust as the confidence in another decision-maker's ability to make reliable and effective decisions. Due to its subjective nature, it is a challenge to accurately measure trust between a *trustor* and *trustee*. Existing methods include surveys with the trustor to self-assess trust (Chen and Barnes 2014), reputation-based models to measure trust based on the subjective impressions derived from the trustee's social interactions (Sabater and Sierra 2005) and performance-based models based on the trustee's performance in past interactions (Witkowski et al. 2001). Most of these studies measure trust as a static quantity (as we do in this work) and only few studies evaluate the evolution of trust based on repeated interactions (Chalkiadakis and Boutilier 2012; Hu et al. 2019). Yu et al. (2013) gives an extensive review of existing approaches to model trust in multi-agent systems.

The literature on trust generally reports a positive relationship between team performance and average team-level trust as a static quantity (Jong et al. 2016). However, a quantitative analysis of the implications of trust between team members when there are sources of uncertainty in the decision-making is missing in the literature.

2.2 Problem Coordination in Design Optimization

Problem coordination has been identified as a critical factor for effective decisions in multiple research fields, including design (Gyory et al. 2019), optimization (Allison et al. 2009), and organizations (Gokpinar et al. 2010). The literature on decomposition-based design optimization has discussed problem coordination as a mathematical problem to iteratively guide the solution of multiple subproblems in order to achieve system optimality and consistency across the solutions of subproblems (Papalambros and Wilde 2017). Several coordination strategies have been proposed to date, including single-level formulations such as multidisciplinary feasible and individual disciplinary feasible coordination (Martins and Lambe 2013), and multi-level formulations such as collaborative optimization (Balling and Rawlings 2000) and analytical target cascading (ATC) (Kim et al. 2003). These strategies interface the subproblems solved by different optimization methods to manage the coupled decisions that affect the solution of other subproblems.

Existing coordination strategies assume that the individual optimizers can solve their own subproblems accurately and the information coming from these solvers are reliable. Incorporating trust mechanisms in these coordination strategies for problems with inadequate information to make reliable decisions has not been studied in the literature. In this work, with a notional case study, we show that trust is a necessary factor that impacts overall performance and hence should be incorporated into the coordination strategy.

3 Methodology

This section presents an approach to quantitatively model a static trust relationship in a computational team that is coordinated with a decomposition-based design optimization architecture. In this architecture, team consists of design agents responsible for solving a portion of the system-level problem and a coordination agent responsible for managing the couplings. We represent the design decisions of the computational agents with deterministic optimization when there is no uncertainty in the models used for decision making and with Bayesian optimization when these models contain uncertainty. We model trust as a weighting factor in the objective function of the coordination agent.

3.1 Generalized Approach

This study uses ATC (Kim et al. 2003) a commonly used coordination strategy from literature to emulate a team-based decision-making scenario. ATC models the subproblems that each agent is responsible for as separate optimization problems with individual objectives, variables, and constraints. The coupled variables among subproblems are independently decided in each coupled subproblem. In order to ensure consistency, ATC defines an additional variable for each coupling as a target for these subproblems and penalizes the deviation from these targets with an extra term added to the objective. Through multiple iterations, ATC guides the subproblems to reach agreements on the values of these coupled variables. The consistency penalty is formulated as an augmented Lagrangian term as follows:

$$\phi(\mathbf{r} - \mathbf{t}) = \mathbf{v}^T \cdot (\mathbf{r} - \mathbf{t}) - ||\mathbf{w} \circ (\mathbf{r} - \mathbf{t})||^2, \tag{1}$$

where $\mathbf{v}$ and $\mathbf{w}$ are penalty weights, $\mathbf{r}$ and $\mathbf{t}$ are subsystem response and targets, respectively, and $\circ$ is the Hadamard product. The penalty weights for each coupled variable are updated through iterations and determine the importance of the deviation from the targets.

In this study, we consider the information sources in some of the subsystems to be uncertain and model the decision making with these sources with Bayesian optimization. We specifically use efficient global optimization (EGO) (Jones et al. 1998) which models an unknown objective function with a Gaussian process. EGO adaptively samples the design space and using the information obtained these samples, it builds a response surface model for the objective. It then creates an expected improvement function $E[I(\mathbf{x})]$ to maximize by combining the existing knowledge of the objective function with the uncertainty as follows:

$$E[I(\mathbf{x})] = \left(f_{\min} - \widehat{f}\right)\Phi\left(\frac{f_{\min} - \widehat{f}}{s}\right) + s\varphi\left(\frac{f_{\min} - \widehat{f}}{s}\right), \tag{2}$$

where $f_{\min}$ is the current best objective function value based on previous samples, $\widehat{f}$ and s are the predicted value of the objective function and its standard error based on the response surface model, and φ and Φ are the standard normal probability density and cumulative distribution functions, respectively. We use this model to determine the best decision to make in every iteration for the subsystems that do not have the full knowledge of the model.

When decisions are based on uncertain sources, trust becomes a critical factor for the effectiveness of the system-level decisions. We particularly focus on the trust of the coordination agent to the design agents and assume that the more coordination agent is confident with the decisions of a design agent, the more importance is placed on the response coming from that agent. Thus, we can represent the amount of trust that the coordination agent has on another agent with an additional weighting factor on the consistency penalty in the objective function. A larger weight corresponds to a higher amount of trust.

3.2 Illustrative Example

In this section, we discuss a notional example to illustrate the application of the trust model in problem coordination. This example uses an analytical function to be minimized as a proxy of a team-level design task and decomposes this task into two coupled subproblems to be solved by individual computational design agents. One, the subproblem is assumed to be deterministic for simplicity, and the other subproblem is modeled with an uncertain objective function. These problems are coordinated with ATC. Branin function is one of the commonly used mathematical functions to test the performance of global optimization methods (Dixon et al. 1978). This function is expressed as follows.

$$f(\mathbf{x}) = a\left(x_2 - bx_1^2 + cx_1 - d\right)^2 + p(1-t)\cos(x_1) + p \tag{3}$$

A minimum value is usually searched for $x_1 \in [-5, 10]$ and $x_2 \in [0, 15]$. The function has three global minima with $f(\mathbf{x}_*) = 0.398$ at $\mathbf{x}_* = \{(-\pi, 12.275), (\pi, 2.275), (9.425, 2.475)\}$ for $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $p = 10$ and $t = 1/(8\pi)$.

We partition this problem into two subproblems and define a coordination problem to model teamwork as shown in Fig. 1. In the first subproblem, Agent 1 has the following local objective,

$$f_1(x_{11}, x_{12}) = \left(a\left(x_{12} - bx_{11}^2 + cx_{11} - d\right)^2\right), \tag{4}$$

and solves the following deterministic optimization problem with two variables:

$$\begin{aligned} &\min_{x_{11}, x_{12}} f_{11}(x_{11}, x_{12}) + \phi(t_{11} - x_{11}) \\ &-5 \le x_{11} \le 10,\ 0 \le x_{12} \le 15 \end{aligned} \tag{5}$$

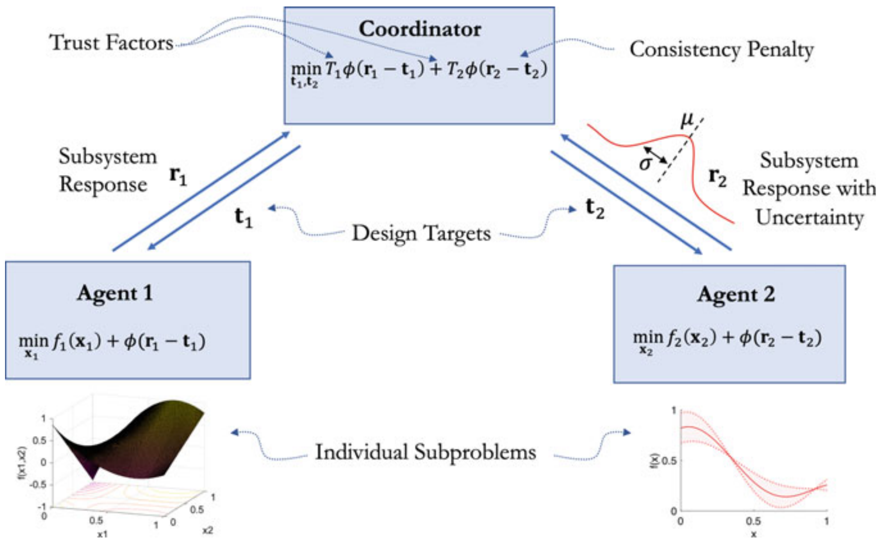


Fig. 1 Overview of the problem coordination under uncertain subsystem responses

where x_{11} and x_{12} are the design variables defined as the local copies of the system-level variables x_1 and x_2, and t_{11} is the design target for x_{11}. The second subproblem for Agent 2 has the following local objective,

$$f_2(x_{21}) = (p(1 - t)\cos(x_{21}) + p), \tag{6}$$

but does not have a full knowledge of this model. It follows an EGO approach to maximize the expected improvement, $E[I(x_{21})]$ based on a Gaussian process model of the objective function.

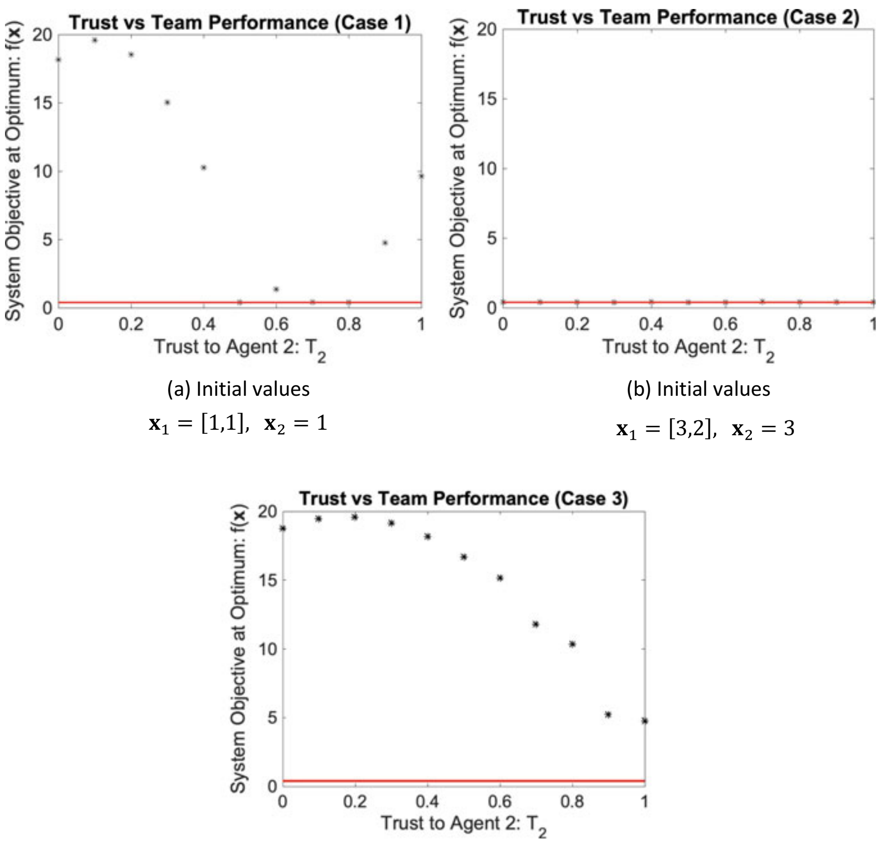
$$\min_{x_{21}} E[I(x_{21})] + \phi(t_{21} - x_{21}), \quad -5 \leq x_{21} \leq 10 \tag{7}$$

where x_{12} is the local copy of the system-level variable x_1, and t_{12} is the design target for x_{12}. The problem of the coordination agent is the following.

$$\min_{t_{11},t_{21}} T_1\phi(t_{11} - x_{11}) + T_2\phi(t_{21} - x_{21}), \quad -5 \leq t_{11}, t_{21} \leq 10 \tag{8}$$

where T_1 and T_2 are the weighting terms in [0, 1] representing the amount of trust to each of the design agents. This study is particularly interested in studying trust to Agent 2 that makes decisions using a probabilistic model. $T_2 = 0$ corresponds to no trust to Agent 2 which lead the coordination agent to ignore the responses coming from this agent, and the problem converges to the local solution of the first subproblem. $T_2 = 1$ represents a case where the responses from this agent are fully considered in the team coordination.

Subproblems of the design agents and the coordination agent can be solved with the Quasi-Newton method. We solve these problems by varying the level of trust to Agent 2 starting from three different initial points and obtain the results shown in Fig. 2. Different initial points for Agent 1 determines the local solution to converge to while they can be interpreted as the problem difficulty from the perspective of Agent 2. As the initial point moves away from the optimal solution, the model uncertainty may lead Agent 2 to an incorrect optimum.



(a) Initial values $\mathbf{x}_1 = [1,1],\ \mathbf{x}_2 = 1$

(b) Initial values $\mathbf{x}_1 = [3,2],\ \mathbf{x}_2 = 3$

(c) Initial values $\mathbf{x}_1 = [6,6],\ \mathbf{x}_2 = 8$

Fig. 2 Comparison of team performance based on varying levels of trust to second design agent. (Solid lines represent the theoretical best performance)

The results in Fig. 2a show that a low level of trust to Agent 2 leads to a low team performance since the local solution of Agent 1 is far away from the system-level solution. While trusting Agent 2 gradually improves the solution, it starts to hurt the team performance when trust in the decisions of Agent 2 is high. This is an example of a case where an optimum trust level to an agent is neither at the lower end nor at the higher end of the spectrum. The results in Fig. 2b show that trust level does not impact the system-level solution since the initial point is very close to the true solution. This is a case that represents an easy design task where trust does not play a critical role in the solution. Figure 2c shows a case where trust in Agent 2 improves the system-level solution. This case is similar to Fig. 2a where the local solution of Agent 1 is a bad solution for the team but the model uncertainties in Agent 2 prevents the team from finding the true optimum.

ATC can identify the global solution of this problem for any of these initial points and for any non-zero trust level when Agent 2 also uses deterministic optimization with the actual model rather than Bayesian optimization. These results show that an optimum level of trust should be sought in order to maximize the team performance when the decisions of the team members are not completely reliable.

4 Conclusions

In this work, we presented a notional case study to quantitatively assess the impact of trust in the coordination of a computational design team on the overall teaming performance when the design decisions in the team are based on information sources with uncertainty. The results showed that trust in an agent should be considered as a design variable and should be tuned to maximize the team performance. Even though these results are obtained from a notional problem, the approach can potentially be generalized to more practical problems in a future study to analyze the relationships among trust, team coordination, and teaming performance. A generalized study can also provide useful insights as to how future human–computer teams should be structured and coordinated with proper trust relationships.

Our analysis in this study is based on a static notion of trust whereas in reality trust evolves over time. Modeling trust as a dynamic quantity is an interesting research problem left to a future study. Also, in this study, we only showed a theoretical relationship between trust and team performance without suggesting a strategy to adjust trust in real time. It is necessary to develop systematic methods to determine a proper level trust for satisfactory teaming performance based on the existing information for the performance history and prior knowledge of the team member capabilities. Finally, the study is purely based on computational teams where trust is modeled as a mathematical quantity. While this approach can be useful to model trust mechanisms for future synthetic team members equipped with artificial intelligence, modeling trust based on actual human factors is critical to understand the dynamics of a real-world human–computer teaming.

References

Allison, J. T., Kokkolaras, M., & Papalambros, P. Y. (2009). Optimal partitioning and coordination decisions in decomposition-based design optimization. *Journal of Mechanical Design, 131,* 081008. https://doi.org/10.1115/1.3178729.

Balling, R., & Rawlings, M. R. (2000). Collaborative optimization with disciplinary conceptual design. *Structural and Multidisciplinary Optimization, 20,* 232–241. https://doi.org/10.1007/s001580050151.

Burton, R. M., Obel, B., & DeSanctis, G. (2011). *Organizational design.* Cambridge: Cambridge University Press.

Chalkiadakis, G., & Boutilier, C. (2012). Sequentially optimal repeated coalition formation under uncertainty. *Autonomous Agents and Multi Agent Systems, 24,* 441–484. https://doi.org/10.1007/s10458-010-9157-y.

Chen, J. Y. C. C., & Barnes, M. J. (2014). Human-agent teaming for multirobot control: A review of human factors issues. *IEEE Transactions Human-Machine Systems, 44,* 13–29. https://doi.org/10.1109/THMS.2013.2293535.

De Jong, B. A., Dirks, K. T., & Gillespie, N. (2016). Trust and team performance: A meta-analysis of main effects, moderators, and covariates. *Journal of Applied Psychology, 101,* 1134–1150. https://doi.org/10.1037/apl0000110.

Dixon, L. C. W., & Szego, G. P. (1978). The global optimisation problem: An introduction. In L. C. W. Dixon & G. P. Szego (Eds.), *Towards global optimisation* (pp. 1–15). Amsterdam, Amsterdam: North Holland.

Gokpinar, B., Hopp, W. J., & Iravani, S. M. R. (2010). The impact of misalignment of organizational structure and product architecture on quality in complex product development. *Management Science, 56,* 468–484. https://doi.org/10.1287/mnsc.1090.1117.

Gyory, J. T., Cagan, J., & Kotovsky, K. (2019). Are you better off alone? Mitigating the underperformance of engineering teams during conceptual design through adaptive process management. *Research and Engineering Design, 30,* 85–102. https://doi.org/10.1007/s00163-018-00303-3.

Hu, W.-L., Akash, K., Reid, T., & Jain, N. (2019). Computational modeling of the dynamics of human trust during human-machine interactions. *IEEE Transactions on Human-Machine Systems, 49,* 485–497. https://doi.org/10.1109/THMS.2018.2874188.

Jones, D. R., Schonlau, M., & William, J. (1998). Efficient global optimization of expensive blackbox functions. *Journal of Global Optimization, 13,* 455–492. https://doi.org/10.1023/A:1008306431147.

Kim, H. M., Michelena, N. F., Papalambros, P. Y., & Jiang, T. (2003). Target cascading in optimal system design. *Journal of Mechanical Design, 125,* 474. https://doi.org/10.1115/1.1582501.

Martins, J. R. R. A., & Lambe, A. B. (2013). Multidisciplinary design optimization: A survey of architectures. *The American Institute of Aeronautics and Astronautics, 51,* 2049–2075. https://doi.org/10.2514/1.J051895.

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model Of organizational trust. *Academy of Management Review, 20,* 709–734. https://doi.org/10.5465/amr.1995.9508080335.

National Aeronautics and Space Administration. (2008). *NASA systems engineering handbook*. Government Printing Office, Washington, DC: U.S.

Papalambros, P. Y., Wilde, D. J. (2017). *Principles of optimal design.* Cambridge University Press.

Ramchurn, S. D., Huynh, D., & Jennings, N. R. (2004). Trust in multi-agent systems. *The Knowledge Engineering Review, 19,* 1–25. https://doi.org/10.1017/S0269888904000116.

Sabater, J., & Sierra, C. (2005). Review on Computational Trust and Reputation Models. *Artificial Intelligence Review, 24,* 33–60. https://doi.org/10.1007/s10462-004-0041-5.

Witkowski, M., Artikis, A., & Pitt, J. (2001). Experiments in building experiential trust in a society of objective-trust based agents (pp. 111–132).

Yu, H., Shen, Z., Leung, C., Miao, C., & Lesser, V. R. (2013). A survey of multi-agent trust management systems. *IEEE Access, 1,* 35–50. https://doi.org/10.1109/ACCESS.2013.2259892.

Style Similarity as Feedback for Product Design

Mathew Schwartz, Tomer Weiss, Esra Ataer-Cansizoglu, and Jae-Woo Choi

1 Introduction

Interior designers and architects are experienced in creating cohesive styles of an interior space. However, the cost of such a professional is prohibitive for most people. Such professionals use intuition, experience, rules in composition, and color theory to drive decision making, along with modes of visual communication such as drawing and 3D rendering. For the typical consumer, interior design and home decoration are a balance of personal taste and guesswork. One of the more difficult barriers to overcome on the general consumer side is the visualization of multiple products within their context (i.e., a room all products will reside in). Similarly, the sorting of massive online catalogs of products and matching these products in a desirable way can be overwhelming.

One sorting approach is through using a set of attributes and labels assigned to products in the catalog. Through such attributes, item-based filtering can be applied, in which items sharing certain features, such as color, are recommended to a user. While this technique is a standard industry practice, it is limited in usefulness for large datasets containing hundreds of thousands of products. Additionally, certain

M. Schwartz (✉) · T. Weiss
New Jersey Institute of Technology, Newark, NJ, USA
e-mail: cadop@umich.edu

T. Weiss
e-mail: tweiss@njit.edu

E. Ataer-Cansizoglu
Facebook, Boston, MA, USA
e-mail: cansizoglu@ieee.org

J.-W. Choi
Wayfair, Boston, MA, USA
e-mail: jchoi@wayfair.com

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_3

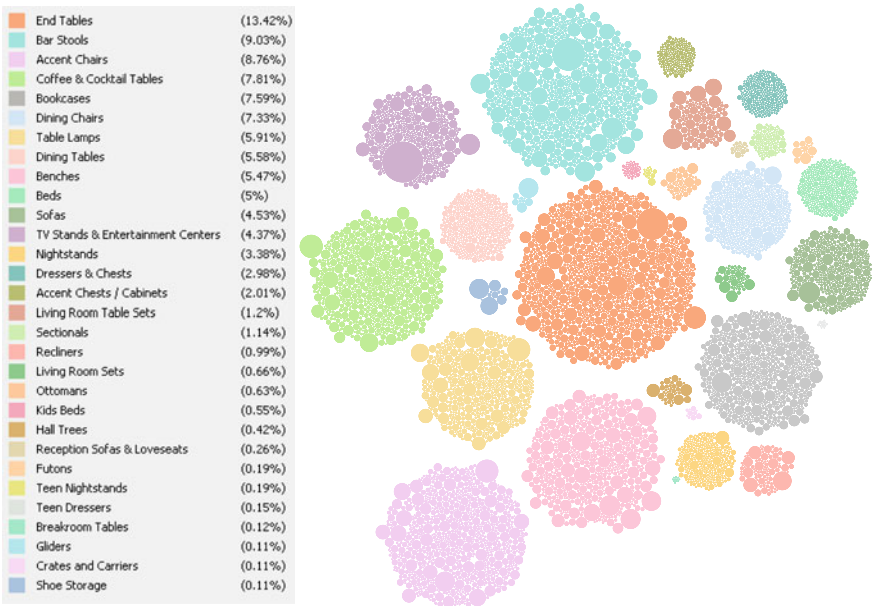


Fig. 1 The nodes of the reduced dataset. Color corresponds to the product group type and size corresponds to the degree of the node

attributes are more abstract and difficult to ascertain, requiring multiple ratings and labels by experts–some of which might not agree on such subjective matter. One such attribute is the style of a given space, which includes multiple components such as: fabric, color scheme, material, furniture style and flooring. In this work, we use such components as the basis of four major room styles: *modern, traditional, cottage, coastal*, that are popular within an interior product e-commerce website (Wayfair). We elaborate on descriptions of these styles in Table 1.

An emerging trend in the e-commerce space is to use image-based analysis for identifying similarities of products. This is useful in online situations where products must have an image (or 3D rendering) for customer display, while not being fully labeled with information describing subjective style-related attributes. In this paper, we propose a synergy between the methods used for recommendation algorithms to a consumer and the design of the products to be recommended. Background and implementation of past work is presented, along with an in-depth discussion on the resulting analysis of our dataset. This dataset is of images labeled with product Stock Keeping Units (SKUs) and similarly, associated with a product group (Fig. 1) from the Wayfair company. By organizing data into a network structure, we are able to visualize and explore connectivity and relationships between various attributes and types. Finally, we present a flowchart for how such image-based analysis of style similarities can be integrated within a product designers workflow, in a not too dissimilar way to that of a consumer browsing online.

2 Related Work

Understanding a user's preferences lies at the heart of many e-commerce websites (Ataer-Cansizoglu et al. 2019; Sachidanandan et al. 2019; Jing et al. 2015). An accurate representation of customer's style preferences enables–better product recommendations and a more personalized shopping experience. The gap in matching an individuals preference with a product is in the extraction and categorization of style-related attributes of a product. Researchers have proposed multiple quantitative methods for understanding (i.e., extracting) subjective attributes related to aesthetics and fashionability that compose a style (Hsiao et al. 2017; Schifanella et al. 2015; Dhar et al. 2011; Simo-Serra et al. 2015; Pan et al. 2019; Lun et al. 2015 with a few that focus on interiors Pan et al. 2019; Lun et al. 2015; Weiss et al. 2018).

Quantitative methods for style recognition are typically data-driven and employ a variety of machine learning techniques. Recently, neural networks, and specifically deep neural networks, have become paramount for automatically assessing images (Goodfellow et al. 2016) and in general are effective for image recognition tasks (Simonyan and Zisserman 2014). A large body of work exists on the categorization of images using such deep neural networks. To find an image's category, a deep neural network learns a lower dimensional representation, which we refer to as style embeddings (See Sect. 3), based on labeled data. These embeddings represent an image's category in an abstract category space, such as weightings within various style types. Distances between embeddings representing different images can then be evaluated to find whether they are similar (Hoffer and Ailon 2015). The two most frequently used data types used by researchers to learn style and visual compatibility are images and/or 3D models.

Using 3D models, Hu et al. (2017) presents a method for discovering elements that characterize a given style, where the elements are co-located across different 3D models associated with such style. Both Liu et al. (2015), Lun et al. (2015) propose to learn 3D model style based on perceptual geometric elements. They employ crowd-sourcing to quantify the different geometric style components. Building on this work, Lim and colleagues Lim et al. (2016) use neural networks to directly identify the style of 3D models, without geometric features, by extracting images of the target 3D model. While the Wayfair e-commerce data consists of 3D models, we focus on the more general use case of image data for determining style similarity. Using images, Karayev et al. (2014) learn the style of paintings and common photographs using linear classifiers. Thomas and Kovashka (2016) use style to identify a photograph's artist, which could have implications for brand identity. Considering a products connection to other products and within societal context, Simo-Serra and Ishikawa (2016) propose to predict an outfits' fashionability using neural networks and Gu et al. (2017) identify fashion trends from street photos. Similar to our work, but focusing directly on product images rather than overall style, (Bell 2015) used deep networks for a visual search system that can aid in interior design through product image similarity.

For design-based companies, understanding consumer interests and brand recognition is strongly linked to visual style attributes. Studies into quantifying design aesthetics for consumer decisions have provided valuable insights through eye-tracking and interactive modeling (Hyun et al. 2017). As elaborated by Burnap (2016), the importance of brand recognition, and the difficulty in balancing with design freedom, is imperative for companies in brand-conscious spaces. While that work was within the automotive industry, cohesion between various single products for a complete style in an interior space is equally important. This cohesion among products as seen by a consumer should be achieved on both the e-commerce side–through intuitive and appropriate recommendations to a consumer; as well as the product design side–by varied designs useable in both different multiple product groupings (a range of styles) and along with other products (including with competing brands).

3 Learning Image Style with Neural Networks

Our work relies on deep neural networks to learn furniture styles through images. Training images were collected in a previous study (Ataer-Cansizoglu et al. 2019; Yildiz et al. 2020) and each was labeled with a designated style by multiple experts (Sect. 3.1). Our neural network is trained to estimate an approximate style of an image, and in doing so, provides a comparative stylistic differences between the images (Sect. 3.2). This stylistic difference is determined by a 16-dimensional layer in the neural network which we denote as *style embeddings*. Given such embeddings, we are able to retrieve similar style images (Sect. 3.3), and by doing so, recommend similar-style products. For a more detailed explanation than the following overview, please refer to the previous work, done in conjunction with Wayfair's Data Science team (Ataer-Cansizoglu et al. 2019; Yildiz et al. 2020).

3.1 Labeling Image Style

Our initial dataset contains 672,000 images of interior scenes and products collected from various sources: (i) staging a scene in a physical room, and then capturing it with a camera, (ii) captured from a virtual scene, curated by a 3D artist, and (iii) scraped from third-parties. However, due to noisy image metadata, an image's origin source might not always be known. Each image is designated with an interior style label which includes: modern, traditional, cottage, coastal (Fig. 2), where each style is described with certain criteria about fabric, color scheme, material, furniture style, and flooring (Table 1).

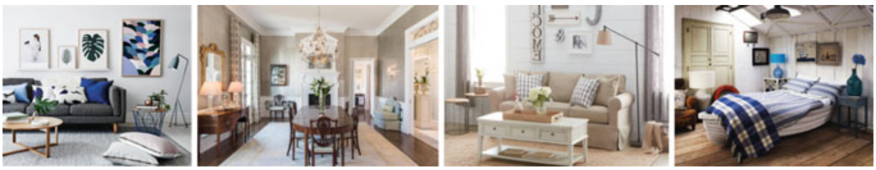
Style labeling is a subjective task prone to noise and variation among multiple people, including stylists. To alleviate discrepancies in dataset labeling, 10 independent experts were used to designate a style label to images. All data, including discrepancies between the expert labels, were recorded, leading to a noisy dataset of

Table 1 Descriptions of four major room styles: modern, traditional, cottage, coastal. Major style features include fabric, color, material, furniture, and flooring

	Modern	Traditional	Cottage	Coastal
Fabric	Heavy texture, leathers, linens	Damask or jacquard, velvet or silk, chintz or florals	Soft florals, linen, checks and gingham, toile	Linen, stripes, nautical
Color Scheme	Muted solids in neutrals, grays and blacks	Blue, dark red, hunter green and brown	Muted blues, pinks, reds and greens, white, pale yellows, soft greens	Blue, white, red, green
Furniture	Sleek, low to the ground, clean lines, straight legs on base	Dark wood, gold accents, antique	Slightly distressed, vintage inspired,skirted sofas or chairs, feminine accents, wooden signs	Whitewashed, distressed, beadboard accents, bamboo and rattan
Material	Mixed	Marble, gold, cherry or mahogany wood	White washed or cherry wood, straw baskets and worn metals	Reclaimed or painted wood, seeded glass or beach glass, beach wood
Flooring	Stripes or natural fiber rugs such as jute or sisal	Ornately patterned carpets	Braided cotton, soft floral or checked rugs	Stripes or woven, seascape prints, sisal or jute

labels–a common issue when training a neural network–as it is prone to inaccurate estimations. Such challenges were overcome by having the neural network compare images within the same dataset and their attributed labels from multiple experts. This assumes the existence of tangible differences between each label that knowledgeable experts may discern between, similar to the methods described in Bradley (1952).

A new set of comparison labels were generated based on the expert-determined style labels. During training of the neural network, a randomly selected sample of images were used in which, for each style, and for each pair of images, a comparison



Modern Traditional Cottage Coastal

Fig. 2 Examples of major room styles label is generated with respect to the relative order of the number of style labels each image receives. Formally, for a given style and an image pair, a comparison label will be $+1$ if in that style, the first image has x more expert votes than the second image in the pair, and -1 if it has less votes. Variations of these values of voting thresholds were experimented with, i.e., $x = 1, 2, \ldots$ and any image pair was discarded that fell outside of the defined range.

3.2 Neural Network Configuration

We use an architecture inspired by siamese neural networks (Bromley 1994) that extends the Bradley-Terry (1952), a neural network that learns from comparisons. VGG16 (Simonyan and Zisserman 2014) forms the base of our siamese neural network. The last layer of VGG16 is removed and instead consecutively add two fully connected layers: a 16-dimensional layer, which informs us of an image's style embeddings, and a 4-dimensional layer, which was used to estimate style.

For the model, 80% of images were allocated for training, 10% for validation, and 10% for testing. Thus, images in the training set were not paired with images in the validation or test sets for generating comparison labels. Our style estimation model was trained on comparisons selected uniformly at random over the training dataset. Next, the resulting models were evaluated on the validation set to determine the optimal neural network parameters. This model was implemented with Python and Tensorflow, where training the neural network model took about 5 hours on a NVIDIA Tesla V100 GPU.

3.3 Style Estimation

Since our goal is to accurately estimate an image's style, we preformed several experiments to assess our neural network performance. Each experiment involved random subsets of our image dataset, and different neural network parameter combinations, including thresholding of expert votes for the comparison labels. Such multiple experiments aid in finding the optimal set of parameters that can most accurately estimate the style of a given image.

For estimating an image's style, we use the last layer of the neural network, which has 4 dimensions, each of which provide a probabilistic estimation of an image's style as modern, traditional, cottage, and coastal (Fig. 3). To evaluate the estimation accuracy, we found the maximum estimation values over all predicted style labels. Our model achieved an average style estimation accuracy of 79% for a random image in our dataset. This estimation accuracy varied by style, with 86.9% estimation accuracy for an image labeled as modern, 74.6% for traditional, 71.6% for cottage, and 67.9% for coastal. One possible explanation to the differences in accuracy is that images with style labels such as coastal can also be subjectively

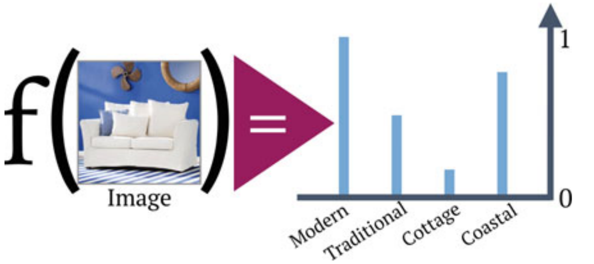


Fig. 3 Our neural network model learns furniture style from images. Given a context image with a target furniture piece (left), we predict its style embeddings

Fig. 4 Product images used for recommendations. Given the furniture image on the left, our model suggests the recommendations on the right

considered as part of another style, such as modern or traditional. Adding more images to our dataset might improve the estimation accuracy per style class.

For recommending similar style images, we use the second to the last layer, which includes 16 dimensions. Given a seed image, we consider this layer as the style embeddings of the image. To evaluate our framework's ability to recommend similar style images, we retrieved the nearest image(s) by using the Euclidean distance from the seed image's embeddings. Fig. 4 demonstrates the 5 closest images to a given seed image, ranked from left to right, ordered by the distance to the seed image. To measure our recommendation performance, we checked if the style labels of the retrieved images match the style of the seed image. Our network's retrieval performance is 73.9% on an randomly sampled image, which is a significant improvement over previous work (Ataer-Cansizoglu et al. 2019).

4 Analysis of Style for Product Design

4.1 Data Processing

We use Networkx (Hagberg et al. 2008) in python for constructing a graph from the relationship data. This graph is then imported to Gephi (Bastian et al. 2009) for visualization. We refer to a given SKU, or SKU and its subsequent options–as a product. By taking the resulting style embeddings comparison for the images of the product dataset from the Wayfair e-commerce site, the graph is constructed by associating products as nodes and the euclidean distance between image style embeddings (Sect. 3.3) as weights. As the distance between images is bi-directional, the result is a weighted undirected graph. To visually analyze such relationships, edge weights are recalculated to $\frac{1}{\text{dist}}$, so that a higher weight corresponds to stylistically stronger similarity. As our data are based on raw e-commerce listings, the staging and advertising of products overlaps between images. To reduce strong connections between images in which the same products exist albeit in a different setting, the edges within the graph of overlapping products (when this is known) are removed.

For a more fine-grained analysis of our dataset, we needed to further reduce the resulting graph. After duplicate data processing, a total of 17,819 products and 2,485,878 weights existed in the graph, with weights ranging from 0.001 to over 100,000, the latter in which images are nearly identical. To remove such noise from the dataset, we cut the graph to include only edges with a weight between 1 and 10. Next, we removed products that were in a group category with less than ten other products. The remaining graph is composed of 8947 nodes and 33,154 edges. Important to keep in mind, the model we use for this paper was designed for style similarity, which as pointed out in Bell (2015), is different than image similarity as an image can be visually similar and hence likely stylistically similar, but similar styles may be different visually.

4.2 Product Relationships

Figure 5 depicts a total of 30 nodes representing a group, where the size of each node relates to its degree. Edges drawn between product groups represent the cumulative weighting of the underlying product connections in the graph. The most common group is the End Table with 1201 products, followed by Bar Stools with 808, and Accent Chairs with 784. Crates and Carriers, Gliders, and Shoe Storage all having 10 nodes.

When viewing the consolidated network, trends among the product groups become apparent. First, we notice the product type itself heavily influences connections to other products as distinct groupings emerge, such as the strongly connected relationship between Bar Stools, Dining Tables, and Dining Chairs. As discussed previously, dataset noise may be a strong factor, since it is likely for similar products to be seen within product images, such as using a Dining Table as staging products for a Dining Chair image. Second, while the total number of End Tables exceeds all other product groups, the sum degree of the Bar Stools group is greater. This relationship may suggest a higher diversity in the styles and relationships a Bar Stool can be recommended for than End Tables.

To further understand group connectivity, we explore relationships between the most connected products in Entertainment Centers, Bar Stools, and Accent Chairs (Fig. 6). We find that Accent Chairs are commonly recommended for other categories (e.g., Entertainment Centers or Bar Stools), yet the most commonly recommended

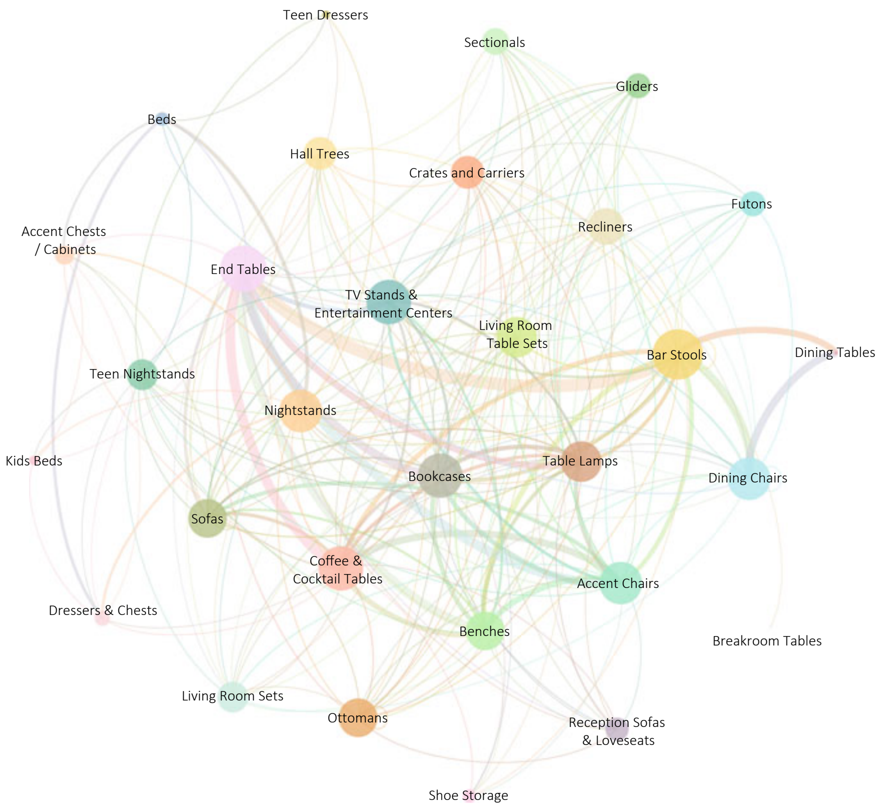
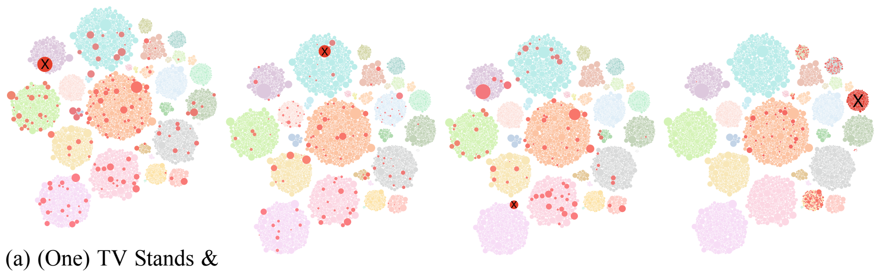


Fig. 5 Connections between the product groups



(a) (One) TV Stands & Entertainment Center (b) (One) Bar Stool (c) (One) Accent Chair (d) (All) Beds

Fig. 6 Similar products of a selected product displayed in red. The groups are colored by the same scheme in Fig. 1. The most frequently recommended product is chosen as the target in the first three images (**a**–**c**), while the last image (**d**) shows the recommendations from all of the products in the Bed group (the red cluster on right side of graph)

Accent Chair has no connection to other products within its own group (i.e., other Accent Chairs). This could be a result of weakly connected products within the same group being filtered out at the initial steps, or interestingly, a result of the style embeddings determining that the features of one Accent Chair do not strongly match those of other Accent Chairs. In this latter case, we interpret this relationship to mean that part of the *style attributes* associated with an Accent Chair is the inherent Accent, in which combining this chair with others would be unlikely. Naturally, the style itself can be similar between these chairs, yet as seen in the example data (Fig. 4), images are a combination of the products within a scene or environment, possibly skewing the relationships to that of a more practical setting in which products exist rather than in isolation. In contrast to the wide-ranging connections between various product groups in Figures 6a–c, Figure 6d highlights (in red) the connections of all products in the Bed group (located on the right side). Such highlighting emphasizes product recommendations are limited to only a few groups: End Tables, Nightstands, Accent Chests/Cabinets, Dressers and Chests, and the smaller categories of Teen Dressers and Teen Nightstands. These results can be interpreted similarly to the Accent Chairs in which the training data lacked examples in which Beds appear in the same image as Bar Stools.

(a) Accent Chair (b) Bar Stool

Fig. 7 Nine most strongly connected recommendations for the most commonly recommended Accent Chair (**a**) and Bar Stool (**b**)

Keeping in mind an error rate of the machine learning algorithm on a more individual image scale, by exploring strong relationships to the most popular products in a group, we gain additional insights as to how a product may contain stylistic attributes determined by our neural network. One key point we found is that product groups found to be strongly connected to an input product generally do not seem to be dominated by color (Fig. 7a). When viewing the strongly connected products in Figure 7b, a theme of large, open, and box-like recommendations seem more common than in Figure 7a.

5 Product Design Loop

In the previous sections, we describe how the neural network works and provide a discussion on some of the results. From this basis, we propose a method for integrating a product design workflow in a manner similar to that of the consumers interaction with the neural network on the e-commerce side. To support future work and continued use of our findings as algorithms in style analysis develop, the abstracted and generalized use case discussed in this section are independent of the speculative and data-exploratory methods we employed in the previous sections.

Current e-commerce systems use a variety of methods to propose alternative and additional products to a consumer; from reviews to collaborative filtering. In Figure 8, the left side illustrates this current practice with the right side demonstrating the *Designer in-the-loop* process proposed here. In both cases, a list of possible products

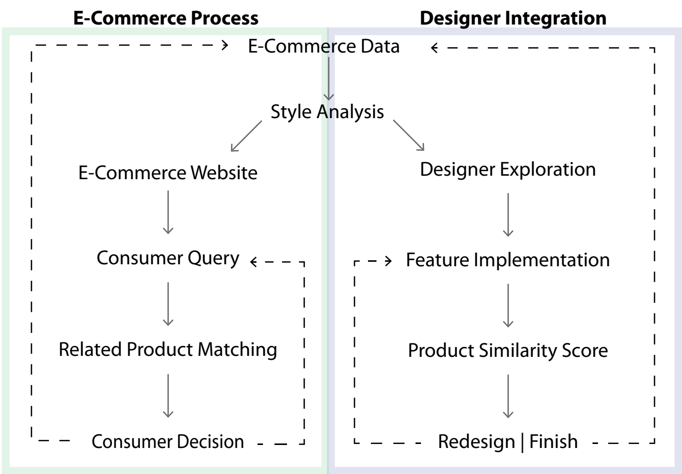


Fig. 8 (Left) E-commerce process for showing related products to a consumer based on their query and shopping preferences. (Right) A mirrored example utilized by the designer to find product similarities of a proposed design

on the e-commerce site are run through an image-based neural network to create a graph of recommendations. On the consumer side, a query is given to the website on a particular product, which results in additional related product matches being displayed. The consumer can then choose from one of these options, after which such choice is updated at an e-commerce backend server (either as a purchase for reducing inventory, or stored for additional filtering methods). Alternatively, the consumer can perform a new query and repeat the process. For the designer, an initial exploration process begins with finding either gaps in the dataset where products are not being connected, or by finding strongly connected features between important/popular products. After implementing such desired features in a design, the rendered image (As the machine learning model is trained on both real and 3D rendered images) can be fed to the algorithm which provides the designer with a product similarity score. From these connections, designer can either redesign and iterate, or finish the design, adding the finalized design to the e-commerce website listings. We propose that this process could enable automated brand cohesion checking as well.

6 Discussion

A main challenge and limitation in the final interpretation of our dataset relates to the uncertainty in the style-based analysis and image matching, as we rely on a subset of the output from the company-provided model. This has posed a few issues with extending our analysis and inferring additional aspects of product style. For one, there are many images that include the same or similar products. While we have a subset of the images labeled with SKUs within them, additional staging props or noise in this list can be meaningful in the recommendations. Similarly, while our data come from one of the largest e-commerce companies, it is by no means a complete set of products and styles. Further implementations and additional product comparison data could provide insights to the addition of external styles. For example, we are unsure of what would happen if a style not included in the training data is given to the model. Additionally, the subjective aspect of *style* can be a hindrance to the machine learning model as culturally and over time the labels and attributes a human perceives as a certain style may change. However, with the increasing accuracy of machine learning models, or with more analysis options for a designer to regenerate style groupings to provide results they more easily understand, our proposed approach may provide a new process for integrating design and big-data analysis for e-commerce use cases.

Wayfair, one of the largest e-commerce websites in the home and decor space, focuses on the huge selection, consistent experience, rich product information, frictionless tools, beautiful content, clear shipping and delivery, and great service to ensure that the experience of shopping for the home is easy and fun. As a market-leading platform, beyond the customer's needs, we focus on the supplier experience by providing the ability to tell a product's story, get to market quickly, and support their business. The capability to find a product design using qualities found in highly recommended products can fill a niche that may currently be underserved in the market and would enable the customers to quickly provide demand signals towards emerging market trends to suppliers. Reducing the overall time in the feedback loop between suppliers and customers in the home and decor space will ensure suppliers have more complete information on market demand and will lead to a revolution in their planning and production process.

Acknowledgements We thank Ilkay Yildiz and Hantian Liu for their preliminary work at Wayfair for the design and experimentation with the neural network style estimation model.

Conflict of Interest The author declares that he has no conflict of interest.

Appendix

See Fig. 9.

Fig. 9 Top-25 most frequently recommended images out of the 37,421 in our sample dataset, with the overall similarity threshold. Such image collection is independent of the products themselves, and exists across all of the original data, including low scoring similarities

References

Ataer-Cansizoglu, E., Liu, H., Weiss, T., Mitra, A., Dholakia, D., Choi, J. W., & Wulin, D. (2019). Room style estimation for style-aware recommendation. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)* (pp. 267–2673). IEEE.

Bastian, M., Heymann, S., & Jacomy, M. (2009). *Gephi: An open source software for exploring and manipulating networks*. http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

Bell, S., & Bala, K. (2015). Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, *34*(4), 98.

Bradley, R. A., & Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika, 39*(3/4), 324–345.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems* (pp. 737–744).

Burnap, A., Hartley, J., Pan, Y., Gonzalez, R., & Papalambros, P. Y. (2016). Balancing design freedom and brand recognition in the evolution of automotive brand styling. *Design Science*, *2*.

Dhar, S., Ordonez, V., & Berg, T. L. (2011). *High level describable attributes for predicting aesthetics and interestingness* (pp. 1657–1664). IEEE.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT press.

Gu, X., Wong, Y., Peng, P., Shou, L., Chen, G., & Kankanhalli, M. S. (2017). Understanding fashion trends from street photos via neighbor-constrained embedding learning. In *Proceedings of the 25th ACM International Conference on Multimedia* (pp. 190–198).

Hagberg, A., Swart, P. S., & Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical Report Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

Hoffer, E., & Ailon, N. (2015). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition* (pp. 84–92). Springer.

Hsiao, W. L., & Grauman, K. (2017). Learning the latent "look": Unsupervised discovery of a style-coherent embedding from fashion images (pp. 4213–4222). IEEE.

Hu, R., Li, W., van Kaick, O., Huang, H., Averkiou, M., Cohen-Or, D., & Zhang, H. (2017). Co-locating style-defining elements on 3D shapes. *ACM Transactions on Graphics, 36*(3), 33:1–33:15.

Hyun, K. H., Lee, J. H., & Kim, M. (2017). The gap between design intent and user response: Identifying typical and novel car design elements among car brands for evaluating visual significance. *Journal of Intelligent Manufacturing*, *28*(7), 1729–1741.

Jing, Y., Liu, D., Kislyuk, D., Zhai, A., Xu, J., Donahue, J., & Tavel, S. (2015). Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1889–1898). ACM.

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., & Winnemoeller, H. (2014). Recognizing image style. In *Proceedings of the British Machine Vision Conference*. BMVA Press. https://doi.org/10.5244/C.28.122.

Lim, I., Gehre, A., & Kobbelt, L. (2016). Identifying style of 3d shapes using deep metric learning. In *Computer Graphics Forum* (Vol. 35, pp. 207–215). Wiley Online Library.

Liu, T., Hertzmann, A., Li, W., & Funkhouser, T. (2015). Style compatibility for 3d furniture models. *ACM Transactions on Graphics (TOG)*, *34*(4), 1–9.

Lun, Z., Kalogerakis, E., & Sheffer, A. (2015). Elements of style: Learning perceptual shape style similarity. *ACM Transactions on graphics (TOG)*, *34*(4), 84.

Pan, T. Y., Dai, Y. Z., Hu, M. C., & Cheng, W. H. (2019). Furniture style compatibility recommendation with cross-class triplet loss. *Multimedia Tools and Applications*, *78*(3), 2645–2665.

Sachidanandan, S., Luong, R., & Joergensen, E. S. (2019). Designer-driven add-to-cart recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (pp. 525–525).

Schifanella, R., Redi, M., & Aiello, L. M. (2015). An image is worth more than a thousand favorites: Surfacing the hidden beauty of flickr pictures. In *International AAAI Conference on Web and Social Media*.

Simo-Serra, E., Fidler, S., Moreno-Noguer, F., & Urtasun, R. (2015). *Neuroaesthetics in fashion: Modeling the perception of fashionability* (pp. 869–877).

Simo-Serra, E., & Ishikawa, H. (2016). Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 298–307).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Thomas, C., & Kovashka, A. (2016). Seeing behind the camera: Identifying the authorship of a photograph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3494–3502).

Weiss, T., Litteneker, A., Duncan, N., Nakada, M., Jiang, C., Yu, L. F., et al. (2018). Fast and scalable position-based layout synthesis. *IEEE Transactions on Visualization and Computer Graphics*.
Yildiz, I., Cansizoglu, E., Liu, H., Golbus, P., Tezcan, O., & Choi, J. W. (2020). Deep ranking for style-aware room recommendation. In *Proceedings of AAAI*. Student Abstract.

Type and Behavior Pattern Analysis of Art Museum Visitors Based on Social Network Analysis

Mi Chang, Taeha Yi, Sukjoo Hong, Po Yan Lai, and Ji-Hyun Lee

1 Introduction

The mission of a museum, as a nonprofit, permanent institution, is to collect, conserve, research, and exhibit the human heritage to educate and entertain the public (Bitgood 1988). In recent decades, digital technologies have opened up new ways for museums to connect the public and benefit from the connections. For example, museums may attract potential visitors by making the collections accessible online. Nevertheless, the benefits come together with challenges. Online access to cultural goods might inversely discourage physical visits. Inappropriate use of technology might also hinder visitors from interacting with the exhibits. As a result, a major struggle of the museums is how to interest visitors with digital means.

From a design perspective, the key is that museums must understand the needs and behaviors of visitors. In museology, common methods to learn about visitors include questionnaires, interviews, experiments, direct observations, and behavioral mapping (International Council of Museums 2017). These approaches are useful to investigate the visitor's subjective feelings and thoughts, as well as how the visitors

M. Chang · T. Yi · S. Hong · P. Y. Lai · J.-H. Lee (✉)
Graduate School of Culture Technology, KAIST, Daejeon 305-701, Republic of Korea
e-mail: jihyunlee@kaist.ac.kr
URL: http://www.ibdsite.com

M. Chang
e-mail: rosechang@kaist.ac.kr

T. Yi
e-mail: yitaeha@kaist.ac.kr

S. Hong
e-mail: sukjoo.hong@kaist.ac.kr

P. Y. Lai
e-mail: pylai@kaist.ac.kr

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_4

interact with space. However, they offer limited insights into the actual visitor–visitor and visitor–museum interactions. These interactions become hard to observe if they happen beyond the museum space.

Social network analysis (SNA) is a set of techniques to study the social structure based on graph theory (Scott 2000). People and their connections are observed, represented, and analyzed as a network. With the network data, the researcher can learn about the community structure and its influences on members' behaviors. The difficulty of SNA exists in data collection as it is costly and laborious. A museum can hardly survey all of its visitors and their private connections. The cost might explain why SNA has long been absent in museum visitor studies, even though it can be a powerful tool to study visitors' characteristics and needs.

Nowadays, the interaction data between visitors and museums become obtainable on social media. Interactions are happening on the social media pages directly managed by the museums. Believing that visitors have diverse characteristics, behaviors, and needs, this study aims at understanding the connections and characteristics of Korean museum visitors by SNA. The research wonders how visitors are different from each other in terms of their characteristics, behaviors, and interests.

To conduct the analysis, we collected data from the official Instagram account of the five most followed Korean museums. We sampled a list of 5000 users from the museum followers. It is reasonable to assume these users might have visited the museum or is a potential visitor. For each of these users, we gathered the list of the accounts they were following and all the hashtags they used. With network analysis, we found that the users can be clustered into six groups. The six groups differ not only in community structure but also in hashtags they used. Analysis of hashtags reveals that the six groups demonstrate different interests and behavioral patterns.

Although the scope of this paper is relatively small compared to SNA studies in other domains, our contribution is twofold. Methodologically, this study demonstrates how SNA techniques can be employed in visitor studies. Meanwhile, our findings provide a new understanding of the Korean museum visitors. This knowledge is practical for actual museum operations. The study can also be the starting point of future research to improve and develop new visitor-centered systems in museums.

The paper is organized as follows. Section 2 reviews existing visitor studies in museums. Section 3 describes the methods and implementation details. Sections 4 and 5 elaborate the results with analysis. Section 6 is the conclusion that discusses the limitations of this study and the future work to be done.

2 Related Work

2.1 Type of Museum Visitor

Contemporary museums have been steadily carrying out research to categorize and characterize visitors. The most common taxonomy is looking at the visitors based on the interaction experience model presented by Falk and Dierking (2018). The model suggested that three interrelated contexts affect an individual's experience. The three contexts are personal, social, and physical contexts. Personal context refers to the unique characteristic of an individual, such as experience, knowledge, interest, motivation, and considerations. Elements of social context include group visits, conversation with museum staff, and personal interpretation of visitors. The physical context consists of the quality of artworks, the atmosphere of space, and the arrangement of the exhibition.

There have also been studies investigating visitors' experiences in detail. Sheng and Chen (2012) analyzed visitors' preferences and demographic factors by their types of experience expectations. They suggested that museum visitors might expect one of the five: easiness and fun, cultural entertainment, personal identification, historical reminiscences, and escapism. Ruiz-Alba et al. (2019) identified two segments, emotional and rational, based on the visitor's experience processing. Their findings showed significant differences between the two segments regarding variables such as visitor satisfaction, loyalty, service experience, emotion, positive disconfirmation, and willingness to pay more.

More and more people nowadays are approaching museums through online channels (Marty 2007; Kotler et al. 2008). Lotina (2014) explored the diversity of participatory activities applied by Latvian museums in online channels, as well as the attitude of museum professionals towards online participation. However, not enough work has been done to investigate the needs of latent visitors online. Therefore, this paper focuses on the social network on these online channels.

2.2 Visitor Studies Utilizing Social Networking Services

Many museums have been using social networking services (SNS), such as Facebook, Twitter, Flickr, YouTube, and Instagram. Most have focused on marketing usage, but not a strategic and tactical level (Hausmann 2012; Chung et al. 2014). For example, Fletcher and Lee (2012) surveyed about the purpose of using SNS of 315 American museums. 60% of American museums used SNS for event listings or posting reminder notices, 45% to post online promotions, 42% to meet new audiences, and 11% for conversational engagement.

To extend the usage of SNS, Vassiliadis and Belenioti (2017) indicated that the museum professionals who manage the SNS need to understand users by monitoring their visitor patterns. Gerrard et al. (2017) investigated the features of visitors who participated in two exhibitions by analyzing the posting on Twitter. Also, Poce et al. (2018) analyzed the accounts of the museum education department on Facebook and Twitter. However, they did not figure out the relationship between accounts and meaningful insights from the collected SNS data. Budge (2017) utilized Instagram to explore visitors' features through the case study by focusing on a particular exhibition. This research showed that visitors prefer to record and remind themselves of the visiting experience on their accounts. However, this paper interpreted visitors with few data (158 posts) without a quantitative approach based on the network analysis. Most visitor studies using SNS data have shown the same insufficiency.

3 Methods and Data Description

3.1 Data Acquisition and Preprocessing

The data acquisition started with Top 5 Korean art museums by their number of followers on Instagram. As shown in Table 1, our research subjects were Daelim Museum, National Museum of Modern and Contemporary Art, Seoul Museum of Art, Ilmin Museum of Art, and Kukje Gallery. We gathered a list of users who are the 1000 most recent followers per museum. Excluding users who set their accounts as private or deleted the accounts during the data collection period, a total of 2838 museum followers remained. To acquire user data from Instagram, we used the Python package *urllib* for working with URLs and *Selenium WebDriver* for automating web application.

For each of the museum followers, we collected the list of accounts they were following. With the lists, we generated the followers' network based on whether the users were following each other. We also retrieved all hashtags used by these museum followers. A hashtag is a type of metadata used on SNS to aid users in searching for similar posts being tagged. Users often put hashtags to explain and indicate the content of their posts. Therefore, the collected hashtags could be regarded as a semantic feature for us to characterize users. During the data preprocessing, we

Table 1 Five most followed Korean art museums on Instagram

Institute	Instagram id	Number of followers
Daelim Museum	@daelimmuseum	124,000
National Museum of Modern and Contemporary Art	@mmcakorea	76,400
Seoul Museum of Art	@seoulmuseumofart	57,600
Ilmin Museum of Art	@ilminmuseumofart	29,700
Kukje Gallery	@kukjegallery	23,500

Data retrieved on March 20, 2019

deleted the combination of symbol characters from the hashtag list when it slices a single word. Emoji characters, on the other hand, were preserved by encoding characters into UTF-8, e.g., 🎨, □, 🎬, 😍, 😁, ❤□, 💖, 🌊, ☕□, 🍕, etc.

3.2 Data Analysis

To compare a network property against another, the degree distribution of grouped networks was analyzed. In the case of a social network, the degree k, and the probability $P(k)$ that degree k will happen, follow a power-law distribution. To turn data points into a linear pattern, the degree k was mapped into a logarithmic scale. We took the natural log of the degree k using *NumPy*, a package that provides high-level mathematical functions to operate on multi-dimensional arrays and matrices for the Python programming language.

With the result that the correlation between the probability of degree k, $P(k)$, and the log of k, $\log(k)$, follows a linear relationship, it was possible to generate the linear regression model to estimate coefficients. The minimum value of $P(k)$, in each dataset, was ignored to minimize the least-squares solution, which is the difference between the true and the predicted value. *Scikit-learn*, a machine learning library, was used to implement the linear regression model. The coefficient of $P(k)$ and $\log(k)$ is represented as the slope on the graph shown in Fig. 1.

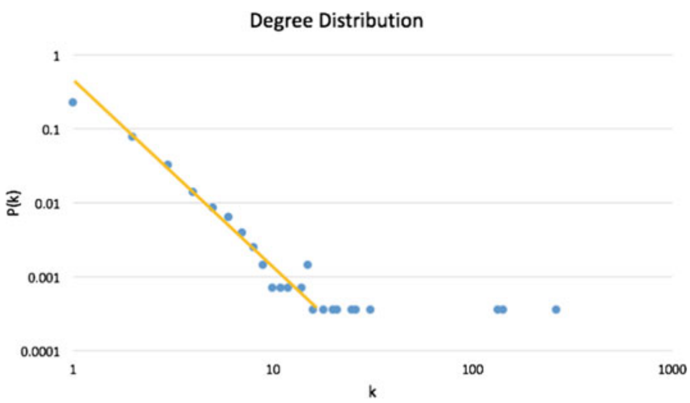


Fig. 1 Fitted linear regression model of $P(k)$ and a degree k in followers' network

4 Whole Network Analysis

4.1 Statistical Analysis of Whole Social Network

We have analyzed the social network of followers who follow five museums in South Korea on the Instagram as mentioned above. Figure 2 shows the relationship of 2838 followers. Originally, it had a total of 5000 nodes with 1000 followers for each museum, but a total of 2838 nodes remained, excluding private accounts and outliers. The outlier means that the user is not following or being followed. Thus, Table 2 shows that 2838 nodes are connected to 1387 directed edges.

The statistical properties of our followers' social network have been calculated in Table 2. The comparison between follower's social network and random network with the same number of average degree, edges, and nodes was created by Gephi.

According to Table 2, compared to the random networks, the followers' network has a shorter diameter and average path length, and the average clustering coefficient has a larger value. This is a more closely connected followers' network, and it has a small-world feature. Also, followers' network has clustering part compared to

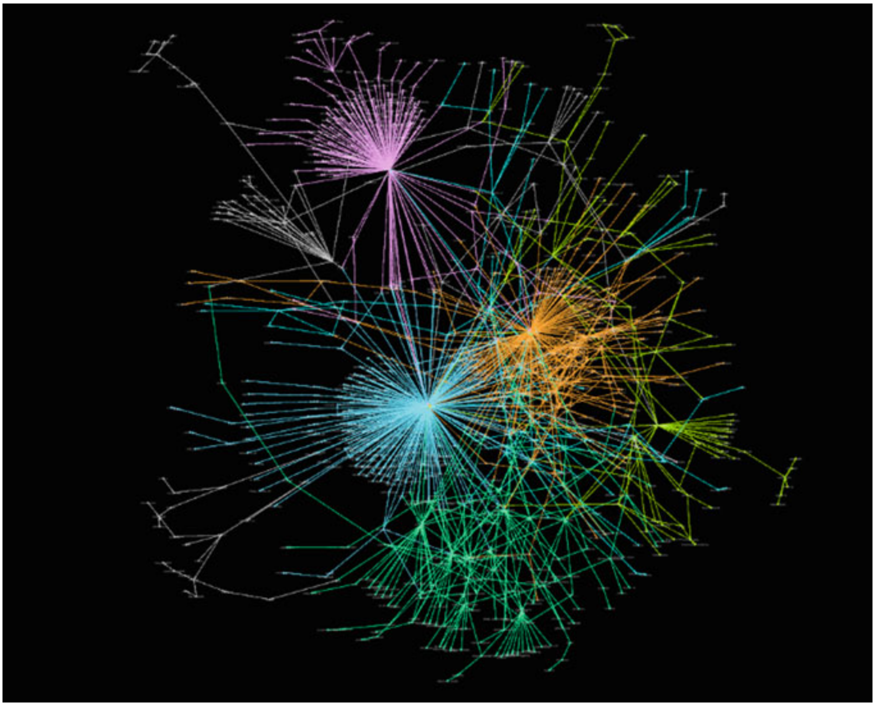


Fig. 2 Social network of followers

Table 2 Statistical properties of followers' social network

Social network type	Node	Edge	Average degree	Diameter	Average path length	Average clustering coefficient	Modularity
Follower's social network	2838	1387	0.977	13	4.262	0.120	0.702
Reference random network	2838	1387	0.977	41	11.361	0.000	–

random network through modularity value. Figure 1 shows that the degree distribution is a power-law distribution. In the followers' network, most users are following the nodes they are interested in or are being followed by nodes that are interested in them.

$$\log P(k) \propto -\gamma \log k \quad (1)$$

As a result, only a small percentage of the nodes are following or followed by a large number of people. According to the Barabasi–Albert model, the slope of the graph in the power-law network is the spectrum of the network as shown in Eq. (1) (Barabasi and Albert 1999).

4.2 Communities Clustered in Network

Followers' network shows the communities that are clustered in the network. The concept for discovering this community is the modularity of the network. The basic equation is shown in (2) below.

$$Q = \frac{1}{2M} \sum_{i,j}^{N} (a_{ij} - \langle t_{ij} \rangle) \delta[C(i), C(j)] \quad (2)$$

M is the total number of connections, N is the total number of nodes, a_{ij} is the connection from node i to j, t_{ij} is the connection from node i to j when randomly connected, and $C(i)$ is the community to which node i belongs. And $\delta[C(i), C(j)]$ means 1 only when $C(i)$ and $C(j)$ are the same community, and 0 when different. Modularity quantifies the number of connections that extend within a community when compared to random connections, assuming a community within the entire network.

As shown in Fig. 3, the modularity class of follower's network is compared with the random network. In the case of a random network, small modularity classes are distributed evenly, but there are specific classes with a large size in the follower's

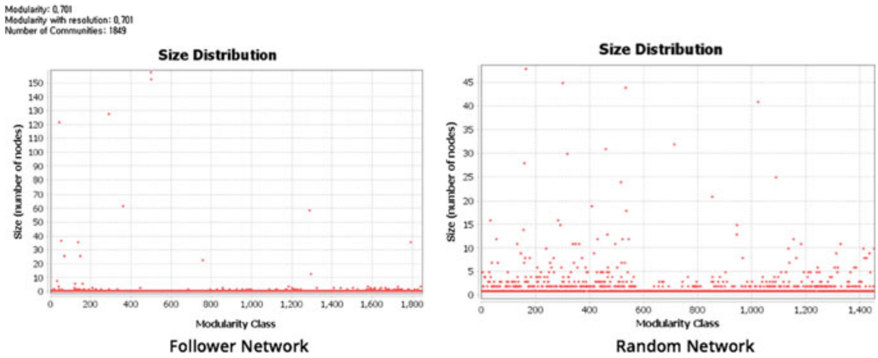


Fig. 3 Degree distribution comparison

network. We divided the class based on the number of nodes 50 and centralization 0.1. The centralization is a value between 0 and 1, normalizing how much network is clustering and quantifying. As a result, a total of 6 high-order groups were analyzed in a follower's network with a centralization greater than 0.1 and a size greater than 50.

4.3 Slope of Groups for Analyzing Hub

The extracted community is called the visitor group, and the network visualization and the slope values in degree distribution are newly calculated for each group. As described above, the slope was calculated using linear regression for each group, and the network visualization was performed as shown in Figs. 4, 5, 6, 7, 8, and 9.

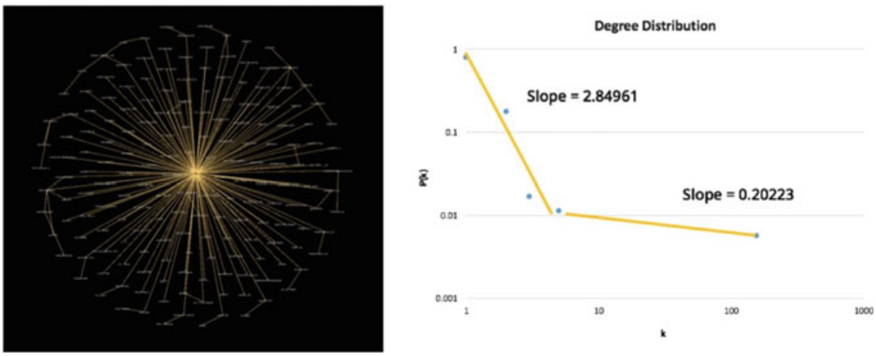


Fig. 4 Group 1

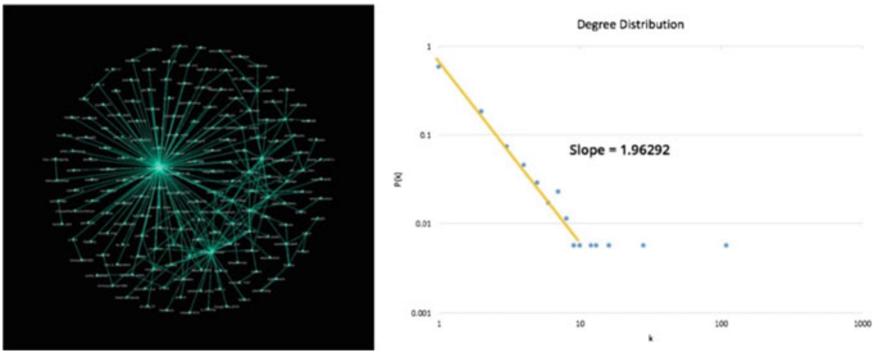


Fig. 5 Group 2

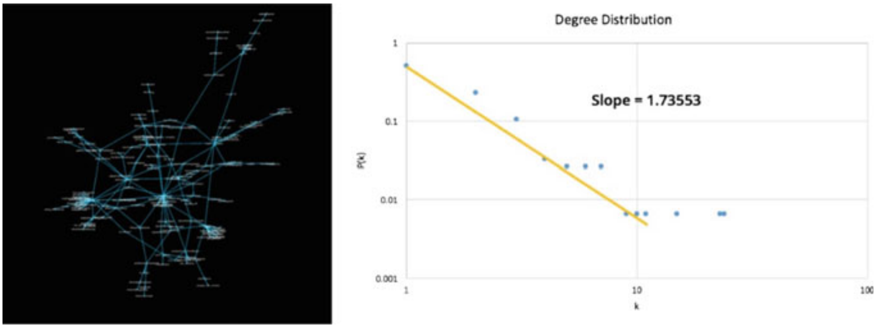


Fig. 6 Group 3

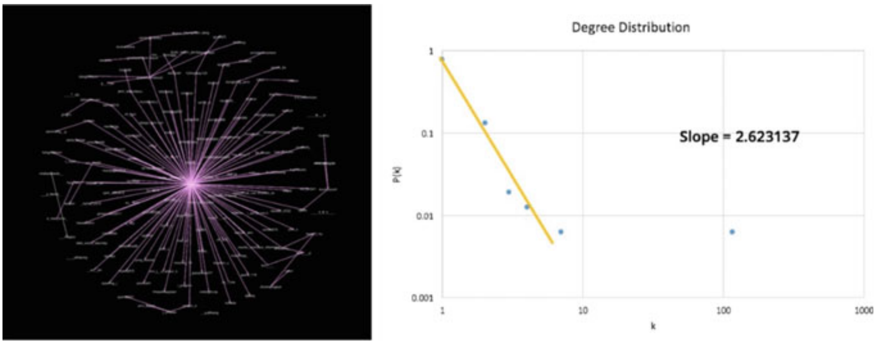


Fig. 7 Group 4

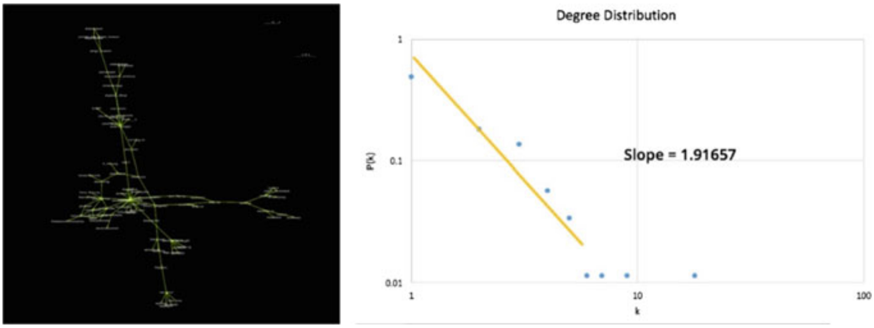


Fig. 8 Group 5

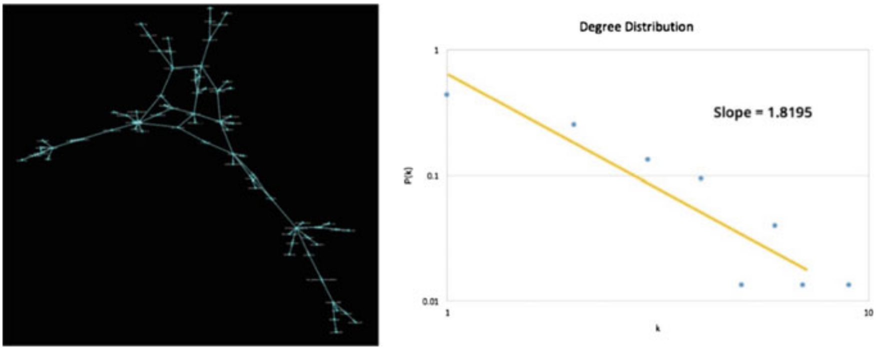


Fig. 9 Group 6

5 Conclusion

Art museums are now returning to the center of visitor, and museums in each country are making efforts to do so. We used data from art museum followers on the SNS for efficient visitor analysis. Based on this analysis, we have shown that the degree distribution of the network is in the form of a power-law, and clustering is performed in six groups through modularity analysis. The statistical analysis was performed based on the groups' network. As shown in Table 3, there were many differences in the centralization and clustering coefficients for each group.

Table 3 shows that Group 1 has the highest centralization and slope is the highest at 2.849 ($\lambda_1 = 2.849$). The average path length is the shortest as 2.3, which is higher than other groups. And the clustering coefficient is the highest at 0.437, so the ratio of triangle is high, and the connectivity between nodes. Group 2 has a much larger number of edges than nodes, but centralization of hubs is not as high as 0.640. The slope is also 1.962 ($\lambda_2 = 1.962$). Group 3 has a slope of 1.736, which is smaller than other groups, and does not seem to be affected by the hub ($\lambda_3 = 1.736$). Each node is subjectively connected based on actual interests, and each node itself has a

Table 3 Statistical properties of groups

	Node	Edge	Average degree	Diameter	Average path length	Average clustering coefficient	Centralization	Slope
Group 1	178	188	1.056	7	2.308	0.437	0.916	2.849
Group 2	176	260	1.477	9	2.936	0.274	0.640	1.962
Group 3	152	190	1.250	11	4.446	0.043	0.239	1.735
Group 4	160	153	0.956	7	2.625	0.000	0.809	2.623
Group 5	88	93	1.057	14	5.272	0.111	0.307	1.916
Group 6	75	84	1.120	16	6.406	0.102	0.154	1.819

higher influence than surrounding influences. Group 4 has fewer edges, clustering coefficient is 0, and triangle is not in network. Only the hub is unilaterally following. Centralization is higher than other groups, and the slope is steep at 2.623 ($\lambda_4 = 2.623$).

Since the average degree is less than 1, the edge is less. However, the hub is overwhelmingly connected and the average path length is 2.6, the second smallest. And Group 5 does not have any other characteristics compared to other groups when viewed in terms of network characteristics. Slope is 1.916, which is less affected by hubs than other groups ($\lambda_5 = 1.916$) Finally, Group 6 has the smallest slope value of 1.819, with the lowest centralization of the network ($\lambda_6 = 1.819$). The diameter is also as high as 16, and the average path length is the longest.

6 Discussion and Future Work

In this paper, we analyze behavior patterns through analysis of network and hashtags of nodes that follow art museums in Korea to analyze art museum visitors. Through the modularity of the whole network, we divided into 6 groups and analyze each network and hash tag. As a result of analysis, each group had distinctly different characteristics, and their behavior also had their own pattern.

This study has analyzed the network and hashtag on the SNS for the art museum visitor; however, this could not apply it to actual users. In the future, we will apply the classification to the actual users, revise them, and apply them to the Lee Ung-no Art Museum in Daejeon. Based on this study, we will analyze visitor type and exhibition information element in real art museum.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. NRF-2019R1A2C1007042), by National Research Foundation of Korea Grant (NRF) funded by the Korean Government (NRF-2019-Global Ph.D. Fellowship Program), and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (R7124-16-0004).

References

Barabasi, A., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286,* 509–512.

Bitgood, S. (1988). An overview of the methodology of visitor studies reliability and validity. *Visitor Behavior, 3*(3), 4–6. Retrieved from https://kora.matrix.msu.edu/files/31/173/1F-AD-122-8-VSA-a0a2h0-a_5730.pdf.

Budge, K. (2017). Objects in focus: Museum visitors and Instagram. *Curator: The Museum Journal, 60*(1), 67–85.

Chung, T. L., Marcketti, S., & Fiore, A. M. (2014). Use of social networking services for marketing art museums. *Museum Management and Curatorship, 29*(2), 188–205.

Falk, J. H., & Dierking, L. D. (2018). *Learning from museums*. Rowman & Littlefield.

Fletcher, A., & Lee, M. J. (2012). Current social media uses and evaluations in American museums. *Museum Management and Curatorship, 27*(5), 505–521.

Gerrard, D., Sykora, M., & Jackson, T. (2017). Social media analytics in museums: Extracting expressions of inspiration. *Museum Management and Curatorship, 32*(3), 232–250.

Hausmann, A. (2012). Creating 'buzz': Opportunities and limitations of social media for arts institutions and their viral marketing. *International Journal of Nonprofit and Voluntary Sector Marketing, 17*(3), 173–182.

International Council of Museums. Statutes. (2017). Retrieved from https://icom.museum/fileadmin/user_upload/pdf/Statuts/2017_ICOM_Statutes_EN_02.pdf.

Kotler, N. G., Kotler, P., & Kotler, W. I. (2008). *Museum marketing and strategy: designing missions, building audiences, generating revenue and resources*. New York, NY: Wiley.

Lotina, L. (2014). Reviewing museum participation in online channels in Latvia. *Museum Management and Curatorship, 29*(3), 280–292.

Marty, P. F. (2007). Museum websites and museum visitors: Before and after the museum visit. *Museum Management and Curatorship, 22*(4), 337–360.

Poce, A., Agrusti, F., & Trasolini, S. S. (2018). Museum education communication in Facebook and Twitter. *Towards Personalized Guidance and Support for Learning, 1,* 372.

Ruiz-Alba, J. L., Nazarian, A., Rodríguez-Molina, M. A., & Andreu, L. (2019). Museum visitors' heterogeneity and experience processing. *International Journal of Hospitality Management, 78,* 131–141. https://doi.org/10.1016/j.ijhm.2018.12.004

Scott, J. P. (2000). *Social network analysis: A handbook* (2nd ed.). SAGE Publications.

Sheng, C. W., & Chen, M. C. (2012). A study of experience expectations of museum visitors. *Tourism Management, 33*(1), 53–60.

Vassiliadis, C. A., & Belenioti, Z. C. (2017). Museums & cultural heritage via social media: an integrated literature review. *Tourismos, 12*(3).

Towards a Useful Grammar Implementation: Beginning to Learn What Designers Want

Andrew I-kang Li and Rudi Stouffs

1 Introduction

Since the very first shape grammar implementation (Gips 1975), the fundamental technical challenge has always been subshape detection (Krishnamurti and Stouffs 1993), the recognition of one shape as being part of another shape, under some allowable transformation. Stouffs and Krishnamurti (2001) identified two additional challenges. The first is generality—the rapid development, adaptation, and maintenance of grammar-based systems—or the ability of users to create and run their own grammars.

The second challenge concerns ways of enabling designers to employ grammatical rules in a manner that does not impede their designing. Tapia (1999) partly addressed this challenge by paying attention to the user experience, in particular visual interaction. In his implementation, users created shapes, not by typing coordinates, but by drawing the shapes directly in a drawing program, that is, as shapes. In shape grammars, shapes, and their component lines and points are domain objects, and now users could manipulate them directly. To apply rules and generate new shapes, they used a stand-alone interpreter, known as GEdit.

The first author, Li (2018), continued this approach by increasing the types of domain objects that users could manipulate directly: not only lines and shapes, the components of shapes, but also shapes and rules, the components of grammars. He used a commercial modeling application, first AutoCad, then Rhinoceros3D (commonly known as Rhino), in which users could draw shapes and rules and handle

A. I. Li (✉)
Kyoto Institute of Technology, Kyoto, Japan
e-mail: andrewli@kit.ac.jp

R. Stouffs
National University of Singapore, Singapore, Singapore

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_5

newly generated shapes. The difference from GEdit was that the whole grammar—the initial shape and the rules—was a Rhino model and could be saved as a Rhino document. Li called this a *whole-grammar* approach, since the whole grammar was available for thinking through direct manipulation.

This provided two advantages. First, users could now directly manipulate all domain objects: rules, shapes, lines, and labeled points. Second, they could now organize rules and shapes in groups to help their thinking, just as they organize icons on their computer desktops. This visual organization is known as secondary notation.

In addition, Rhino offers a wide range of tools, with which users could pre- and postprocess grammatically created shapes. They could, for example, create complicated shapes with scripts, modify those shapes with grammars, and convert those shapes to solid models for 3D printing. Users could now use grammars in a larger workstream of designing and making.

Li's arrangement had a significant disadvantage, however. Like Tapia's, it consisted of two separate applications: one for creating and manipulating shapes, the other for applying rules and calculating new shapes. Where Tapia had GEdit, Li had Grammar Environment, an application developed by Chau et al. (2004).

This split arrangement meant that when users switched tasks, they had to switch applications. And, when switching applications, they also had to transfer shape and rule files. Li provided a library, written in the Python programming language, to facilitate file transfer and similar tasks, but this non-domain work clearly impeded users' ability to focus on domain objects and operations.

We now address this issue by replacing the stand-alone interpreter with one that runs inside Rhino. This new back end is SortalGI, the *sortal* grammar interpreter (Stouffs 2018c). It supports subshape detection and is written in the Python programming language, as are the front end scripts, which have been adapted from the previous version. In this integrated version, all scripts run in Rhino (both v5 and v6 for Windows, and v6 for Mac), and users can do their grammatical work entirely within the Rhino environment.

2 The Front End

Since the front end is set inside the Rhino environment, it is only one of many functionalities already available there. Its purpose is to enable users to use the back end easily. One of the ways it does so is by rendering the components of the grammar—those Rhino objects drawn by the user—parsable for communication to the back end. And for this, the front end needs to use some of Rhino's organizational capabilities, such as layers. At the same time, we want to keep those very capabilities available to users to the largest extent possible. In anticipation of this tension between structure and freedom, we followed a few working guidelines.

First, shapes are composed of Rhino objects, currently line curves and text dots. That is, the objects themselves are the shape; there is not some symbolic representation between the shape and the user. They are in the foreground, available for manipulation by the user, and persist between work sessions as the record of the grammar.

Second, the Rhino workspace is divided between the system and the user. The system needs a place to do its work, and so do users. One way we try to accommodate both system and users is by having the system display rules and all calculated shapes in the positive-*y* half of the three-dimensional virtual workspace. The other half is left for users to use as they will. The other way is by assigning rules and calculated shapes to their own, automatically named layers. This way, the front end can identify rules and calculated shapes, and users can also create and name layers for their own use.

Third, commands are implemented as Python scripts, as in the previous, split version. Users invoke a command by running a script. This is an interim measure; in future versions, commands will be available through more straightforward means, like menu items.

These working guidelines are intended to make users' experiences easier and more productive. The first step for users is to initialize the Rhino document by running the *initialize* script. This prepares both the back end, by initializing its internal representation and rule register, and the front end, by creating a new layer, *Shape 0*, for the initial shape. Shapes on layers named *Shape n* are automatically scrolled so that results are shown chronologically along the positive *y*-axis with the latest result always displayed just above the *xz*-plane.

If the document already contains a grammar, the front end reads the rules into the rule register, and users can resume using the grammar immediately. If the document is new, then it is time for users to create at least an initial shape and a rule. To create the initial shape, users simply draw it with line curves and text dots in the positive-*x*, positive-*y* quadrant of the space (the upper right quadrant of the *xy*-plane in two dimensions) on the layer *Shape 0.*

To create the rule, users draw below the *xz*-plane on any user-named layer. Then they run the *create rule* script, which prompts them to select: (1) the elements of the left shape; (2) the reference point of the left shape; (3) the elements of the right shape (if any); and (4) the reference point of the right shape (if not empty).

The system then draws the rule in the negative-*x*, positive-*y* quadrant on a new layer *Rule 1.* Around each of the left and right shapes is a three-dimensional frame which indicates the shape's coordinate system (Fig. 1). The elements of the two shapes and both frames are locked in a single group for easy selection for rule application. If users draw a second rule, previously created rules will be scrolled away from the *xz*-plane to make space to draw the new rule. Each new rule is entered into the rule register.

Now users have at least one shape and one rule. They run the *apply rule* script, which prompts them to select: (1) the shape; (2) the subshape (optional); and (3) the rule. By selecting a subshape, users can, for example, apply a rule to a single cell in a matrix.

The system calculates all next shapes and, if any, draws them in a row in the positive-*x*, positive-*y* quadrant, each shape on its own layer, and scrolls them—along with all earlier generations of calculated shapes—away from the *xz*-plane (Fig. 2).

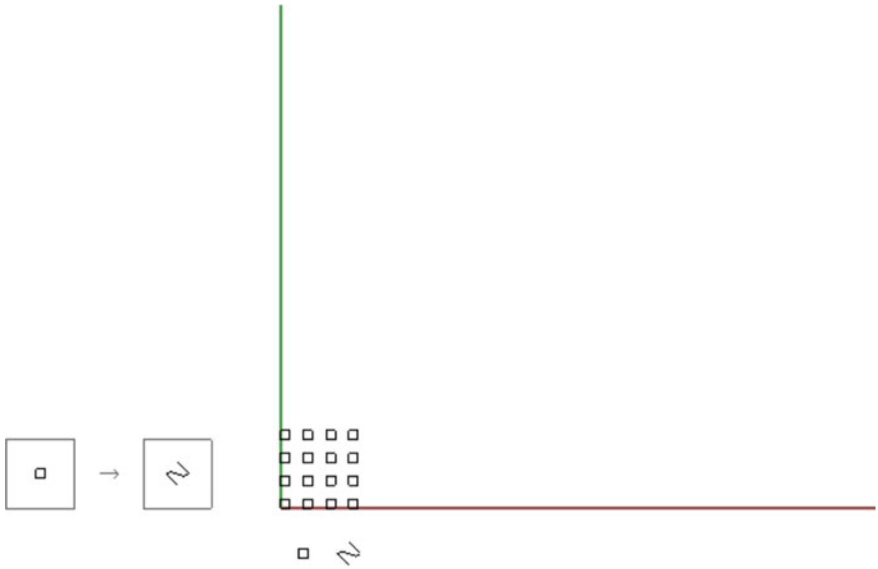


Fig. 1 Two-dimensional (top) view of the workspace, containing a grammar before the first rule application. The rule (with frames around the two shapes) is in the upper left quadrant, while the initial shape (the 16 squares) is in the upper right quadrant. The two shapes in the lower right quadrant are the shapes drawn by the user to define the rule

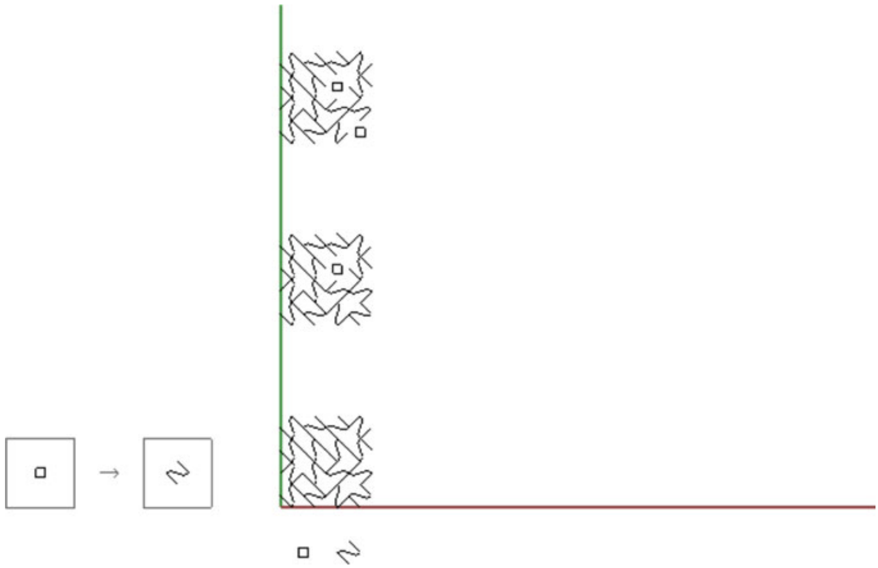


Fig. 2 Two-dimensional (top) view of the workspace after 16 rule applications. Only the last 3 generations are shown here; the first 13 generations are scrolled away from the *x*-axis

3 The Back End

The back end consists of the SortalGI shape grammar interpreter library and API, developed in the Python programming language. Beyond lines and labeled points (text dots in Rhino), SortalGI actually supports a variety of different shape grammar forms, including representations and matching algorithms. SortalGI presents a modular implementation of a generalized shape grammar interpreter, utilizing *sortal* structures as representational building blocks. *Sortal* (representational) structures are hierarchically defined as formal compositions of other, primitive, data structures, denoted *sorts* (Stouffs 2018b). The main compositional operators are a coordinate, disjunctive composition of any number of *sortal* structures, as in a composition of lines and points, and a subordinate, semi-conjunctive composition of a primitive *sort* with any other *sortal* structure under an object–attribute relationship, as in the case of labeled points—points with labels as attributes. Where the fundamental technical challenge is subshape detection, *sortal* grammar formalisms (Stouffs 2012; Stouffs and Krishnamurti 2001) benefit from the fact that every composite structure derives its matching mechanism for subshape detection from the respective matching mechanisms of the component structures. This means that, ultimately, the matching mechanism of any composite *sortal* structure derives from its primitive component *sorts*, solely depending on the respective compositional operators. Having implemented the matching mechanisms for each of the primitive *sorts*, any composition thereof has its matching mechanism implemented as well. In this way, the SortalGI interpreter allows for a broad range of shape grammar formalisms to be supported, including many formalisms found in shape grammar literature.

Beyond its representational flexibility, the SortalGI library includes two alternative matching mechanisms for spatial elements: a nonparametric mechanism matching shapes under similarity transformations (translation, rotation, reflection and uniform scaling) and a parametric-associative mechanism matching shapes under some topological constraints as well as associations of perpendicularity and parallelism. The former recognizes shapes based on similarity, a square matches any square, irrespective of its location, orientation or size. Similar for rectangles of a fixed length-to-width ratio. The latter extends the matching mechanism to polygons of a specific arity. A convex quadrilateral matches any other convex quadrilateral polygon, irrespective of its exact shape, on condition that the former has no perpendicular or parallel edges. Any such perpendicular or parallel edges in the shape to be matched would need to be matched in the target shape.

Note that while the SortalGI library adopts a graph-based representation for parametric-associative shapes, unlike other graph-based implementations (Grasl and Economou 2013; Strobbe et al. 2015; Wortmann 2013), it does not use any subgraph matching algorithm but instead relies on a combinatorial enumeration of potential matches. In general, graph-based, parametric subshape recognition is nonpolynomial, even with a hypothetical, linear time subshape detection algorithm (Wortmann and Stouffs 2018). In comparison, a combinatorial enumeration, searching for k elements within a set of n (distinguishable) elements, yields a tight bound of $O(nk)$.

Depending on the size of k, this bound is exponential in the worst case, while one can use attribute labels to limit the combinatorial explosion.

As such, the SortalGI library supports both parametric-associative and nonparametric shape grammars, including points, line and plane segments, circular and elliptical arcs, quadratic Bezier curves, labels, weights, colors, enumerative values, and (parametric) descriptions, in 2D and 3D. Emergence is naturally supported. Currently, the front end utilizes only the nonparametric mechanism, applied to lines and labeled points. However, it can be extended to include other domain objects as well as the parametric-associative matching mechanism. Nevertheless, even if the SortalGI library is available in its entirety within the Rhino modeling environment, the API provided does limit the extent of geometric and non-geometric element types that are supported, due to the need to graphically visualize the data within Rhino.

This API has been specifically developed to support the integration of the SortalGI library within Rhino. In particular, it not only acts as a programming interface providing access to the underlying functionality, but also supports the conversion of geometric data from Rhino into the SortalGI interpreter and back. At first, the conversion was done from an agnostic description instead of from Rhino's internal object representation. The agnostic description was conceived as a text-based data structure, not unlike the one that was used with Chau et al.'s (2004) engine. The current version of the API no longer accepts the agnostic description and instead relies on Rhino GUIDs referencing Rhino objects. This is possible because the engine runs within Rhino and can query Rhino objects directly using Rhino's Python API.

The API mainly offers functions to create shapes from lists of Rhino GUIDs, to create rules, to determine rule applications and to draw the resulting shapes. The drawing process is an integral part of the conversion from the *sortal* representation back to Rhino GUIDs, in order to generate these GUIDs. However, in order to allow the front end to decide when and where which results should be visualized, the visibility of the resulting Rhino objects is turned off by default.

The library maintains a rule register, and every rule is automatically added to the register. For this reason, rule names must be unique. Rules can be retrieved from the register by name. The library also maintains a shape register although that is entirely voluntary and serves little use in the context here described. Additionally, the library also maintains a register of rule applications both to improve rule application performance, when the same rule is applied to the same shape more than once and to allow for the conception of a derivation tree. There remain many questions as to how such a derivation tree may be accessed and visualized within a platform such as Rhino and, as such, this functionality is currently underdeveloped and unused.

One difficulty encountered has been the precision of geometric information within Rhino. To adjust the precision of computation within the library to the requirements of Rhino, a precision parameter has been created within the SortalGI library that can be set and adjusted through the API.

4 Discussion

Here we discuss our experiences we collected in workshops and a graduate-level course[1] for design students, most of whom had no previous experience of shape grammars.

4.1 The User Experience

We recall that the proximate motivation for adopting an internal back end was to eliminate the mental friction of repeatedly shifting focus and files between applications. The idea was straightforward, but the effect was as consequential as we had anticipated: users simply spent more time on design tasks. As a result, it was easy to observe that users were interested in grammatically generated shapes as objects for their design work, but not as elements in a structured design space. Users accumulated shapes, periodically evaluated them as individuals on their own account, and then culled them without, for instance, trying to preserve derivational sequences.

And if users were not much concerned by the technical niceties of shape grammars, neither were they willing to forsake other design tools for shape grammars. For example, instead of drawing complicated shapes by hand, they wrote Python scripts to create them. They then applied grammar rules to these shapes. In other cases, they used grammars to create what were essentially design diagrams, consisting of zero-width lines. They exported the diagrams to Illustrator, where they assigned (nonzero) widths to the lines and extracted the edges of those lines as cutting paths.

These observations are entirely consistent with Woodbury's (2010) assertion that designers are pragmatic, use whatever tools will help them in their work, and tend to use those tools with less technical sophistication than expert users. This suggests that there is a slight conflict between the authors' interest in domain objects and operations on the one hand and designers' interest in getting the job done on the other. Or perhaps we should say that domain objects and operations are not the same for designers as for the authors, the specialists who created the tool.

For example, we noticed that there were situations in which users were applying the same rule repeatedly to the same shape, when filling in cells in a matrix, for example. This was inefficient, not only because the user had to run the script for each reapplication, but also because the system was redoing many of the same calculations. We wrote a script for applying a rule repeatedly, and the users received it enthusiastically. It was slow, but it freed users from what was to them unnecessary work. The back end offers what are known as *flows*, which are practically composite shape rules and provide a generalized capability for rule application. Users showed us the importance of this capability, and we expect to develop it further.

[1]CAAD Futures workshop, June 2019, Daejeon, South Korea; Ouroborous workshop, August 2019, National Yunlin University of Science and Technology, Yunlin, Taiwan; Advanced Computational Design course, Fall 2018 and Fall 2019, Kyoto Institute of Technology, Kyoto, Japan.

4.2 The Grammarian's Perspective

Similar to the development of the front end interface, the SortalGI library is conceived and developed with shape grammars in mind, rather than simply as a rule engine supporting search and replace. As such, spatial elements are distinguished as either 2D or 3D and as adhering to either a nonparametric or a parametric-associative matching mechanism. Even though it supports multiple grammar forms, the library provides little support for switching between grammar forms. This may not be much of an issue for now, as the interface acts only upon 3D shapes and utilizes only the nonparametric matching mechanism, for now. However, future developments may see the utilization of the parametric-associative matching mechanism as well, possibly requiring the exchange of data between different forms and representations.

Most shape grammars in the literature also consider a single formalism to apply for the entire grammar. Even if multiple formalisms are conceived to be used simultaneously, they are generally used in parallel (Li 2001), and any exchange of information between these parallel representations relies on description rules to include references to other descriptions or shapes. Such exchange is fully supported in SortalGI (Dy and Stouffs 2018; Stouffs 2018a). Duarte (2001) conceives of a discursive grammar that combines two grammars sequentially, a programming grammar generating design briefs based on user and site data and a designing grammar using the design brief(s) to generate designs in a particular style. However, the programming grammar is only a description grammar and any exchange of information between the two grammars is accomplished through description rules operating on the descriptions resulting from the programming grammar.

Acknowledging that a design that originally was conceived in plan or section, or both, might be further elaborated in three dimensions, or the designer might want to use both nonparametric and parametric-associative shape rules intermittently, the SortalGI API does attempt to address these limitations. In particular, it provides support to exchange data between two- and three-dimensional representations, assuming these otherwise correspond to a large degree. Similarly, it supports the exchange of data between representations adhering to the nonparametric and parametric-associative matching mechanisms, again assuming some correspondence between both representations.

4.3 Explorations of Extended Functionality

Beyond its use within the Rhino modeling environment, as presented in this paper, the SortalGI library can be accessed and employed in (at least) two more ways: firstly, within a Python development environment; secondly, as a Rhino/Grasshopper plug-in (Dy and Stouffs 2018), requiring no programming or scripting. These offer access to additional functionality and attempt to address some of the restrictions of the visual front end presented in this paper. For example, the Grasshopper plug-in supports both nonparametric and parametric-associative shape rules to be defined, and for these rules to be applied intermittently. However, it should be noted that the underlying data exchange mechanisms are not entirely general, as it is difficult to establish a general data exchange mechanism to apply between any two *sortal* representations when these representations are unknown in advance and, in the extreme, may not have much in common. We expect the future use of SortalGI to provide more insight to which extent such exchange of data is actually important and desired.

Another matter for further consideration is the relation between the front end and the SortalGI Grasshopper plug-in, which may be considered as two competing interfaces to the SortalGI interpreter, but might also serve as complementary ways of accessing its functionality. Specifically, the integration of the front end with the plug-in could support users in generating complex shapes and rules, while at the same time providing more flexibility in applying rules. Currently, the front end and the plug-in each embody their own initialization setup which are not entirely compatible. Instead, adopting a common *sortal* representation and ensuring a common initialization may allow users to switch back and forth. Besides this integration, user experiences from either or both interfaces may demonstrate best practices and influence their individual development.

In terms of rule application, the plug-in provides four different components that to some extent reflect ways of adopting the front end as outlined before. The first one applies only a single match, which may be selected by index or, otherwise, randomly.

The second one is to conform to the behavior of the front end. It applies all matches in parallel, returning as many results as there are matches and allowing these all to be visualized separately.

The third one may serve to apply a single shape rule repeatedly to fill cells in a matrix. Applying all matches one after another, it ignores the possibility that the actual result from one application may no longer be able to serve as the input for the next application to match. Note that only in the case of an additional rule does this amount to the exact same behavior as repeating the rule application truly sequentially. In the latter case, if any part of the shape is removed as a result of rule application, then this removal will impact subsequent rule applications. This is not the behavior of the third rule application node. Instead, visually, it behaves as if all matches are applied in parallel, combining all the results into a single shape outcome.

Finally, the fourth one can be used to apply a rule repeatedly, although it must be specified how many times. It takes a series of rules as input and applies each rule in sequence, returning all intermediate results as well as the final result. In addition, the plug-in offers a component that does not actually apply the rule, but, instead, returns all the matches. As such, it supports a form of search and extraction based on the left side of the rule. These may then serve as input to one or more rule applications and, as such, improve the efficiency through the application of a divide-and-conquer technique.

5 Conclusion

We presented a general shape grammar implementation that supports subshape detection and handles lines and labeled points in three-dimensional space. Its front and back ends are both set in the CAD application Rhinoceros3D. Informal observations of designers using the implementation suggest that they are more interested in producing designs to work with than in using the more specialized features of shape grammars. Clearly, we need to understand better how designers use grammatical tools. Even though they may not aspire to be grammatical specialists like the present authors, they have much to teach us about how to make grammar more useful to them.

References

Chau, H. H., Chen, X. J., McKay, A., & de Pennington, A. (2004). Evaluation of a 3D shape grammar implementation. In J. S. Gero (Ed.), *Design computing and cognition '04* (pp. 357–376). Dordrecht: Kluwer.

Duarte, J.P. (2001). Customizing mass housing: a discursive grammar for Siza's Malagueira houses. Ph.D. Thesis, MIT.

Dy, B., & Stouffs, R. (2018). Combining geometries and descriptions: A shape grammar plug-in for Grasshopper. In *Proceedings of eCAADe 2018* (Vol. 2, pp. 499–508), Lodz, Poland.

Gips, J. (1975). *Shape grammars and their uses: Artificial perception, shape generation and computer aesthetics*. Basel: Birkhäuser.

Grasl, T., & Economou, A. (2013). From topologies to shapes: Parametric shape grammars implemented by graphs. *Environment and Planning B: Planning and Design, 40*(5), 905–922.

Krishnamurti, R., & Stouffs, R. (1993). Spatial grammars: motivation, comparison and new results. In U. Flemming & S. Van Wyk (Eds.), *CAAD futures '93* (pp. 57–74), North-Holland, Amsterdam.

Li, A. I. (2001). A shape grammar for teaching the architectural style of the *Yingzao fashi*. Ph.D. Thesis, MIT.

Li, A. I. (2018). A whole grammar implementation of shape grammars for designers. *AI EDAM, 32*(2), 200–207.

Stouffs, R. (2012). On shape grammars, color grammars and sortal grammars. In *Proceedings of eCAADe 2012* (Vol. 1, pp. 479–487), Prague.

Stouffs, R. (2018a). Description grammars: Precedents revisited. *Environment and Planning B Urban Analytics and City Science, 45*(1), 124–144.

Stouffs, R. (2018b). Implementation issues of parallel shape grammars. *AI EDAM, 32,* 162–176.

Stouffs, R. (2018c). https://www.sortal.org/downloads/python.html. Last accessed March 7, 2019.

Stouffs, R., & Krishnamurti, R. (2001). Sortal grammars as a framework for exploring grammar formalisms. In *Mathematics and design 2001* (pp. 261–269), Geelong, Australia.

Strobbe, T., Pauwels, P., Verstraeten, R., De Meyer, R., & Van Campenhout, J. (2015). Toward a visual approach in the exploration of shape grammars. *AI EDAM, 29*(4), 503–521.

Tapia, M. (1999). A visual implementation of a shape grammar system. *Environment and Planning B: Planning and Design, 26*(1), 59–73.

Woodbury, R. (2010). *Elements of parametric design*. London: Routledge.

Wortmann, T. (2013). Representing shapes as graphs. Master's thesis, MIT.

Wortmann, T., & Stouffs, R. (2018). Algorithmic complexity of shape grammar implementation. *AI EDAM, 32,* 138–146.

Shape Machine: A Primer for Visual Computation

Athanassios Economou, Tzu-Chieh (Kurt) Hong, Heather Ligler, and James Park

1 Introduction

Imagine working on a paper with your familiar word processor in your computer. As you work, you decide to edit a word only to realize that the word cannot be selected as intended, but only a part of it, few letters in no apparent order. You try to edit another occurrence of the same word in the text, and you realize that this time your selection included some fragments of other words adjacent to the one you started with. Frustrated you erase the complete selected mess determined to retype the correct version of the text, only to realize that there are other letters and fragments of letters underneath the text you deleted! In the end, you decide that the only way to modify the text is to retype everything.

Clearly, this scenario is imaginative, but it is used here as a prompt to illustrate the everyday frustration of designers, engineers, scientists, and all users working with lines in 2D and 3D geometric models. Every designer, engineer, and scientist who works with geometry models knows how difficult is to work with existing files—especially if they are done by others—and how difficult or often impossible, simple operations with lines like the *Find* and *Replace* command discussed above, can be.

This odd situation has of course not been unnoticed. As early as the early seventies, Stiny and Gips called out the need for a uniform representation of shape and over the years Stiny and a growing number of scholars, educators, designers, and scientists relentlessly built a school of thought starting at the Open University and the Royal College of Art at UK, later at UCLA and CMU, and since the mid-nineties at MIT, put together the formidable theoretical formalism of shape grammars with the promise of revolutionizing computer-aided design (CAD) (see, e.g., Stiny 1980, 2006; Krishnamurti 1981; Knight 1994; Earl 1997). Shape grammars' foregrounding

A. Economou (✉) · T.-C. Hong · H. Ligler · J. Park
School of Architecture, College of Design, Georgia Institute of Technology, Atlanta 30332, USA
e-mail: economou@gatech.edu

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_6

of shape rules drawn in a 2D or 3D modeling system over instructions defined in some programming language have provided a robust theory for designers to believe in but nevertheless a formidable challenge to implement. In fact, up-to-date, there are more than 50 shape grammar interpreters that have been designed since the mid-seventies all claiming they can address a particular aspect of shape mapping replacement, the paramount characteristic of this technology (Eloy et al. 2018)—and still none of them has solved conclusively the problem of *Find* and *Replace,* casting thus a shadow of doubt on whether this task is feasible.

The work here takes on this specific problem and proposes a new computational technology, the *Shape Machine*, a new software built for scratch that fundamentally redefines the way shapes are represented, indexed, queried, and operated upon (Hong and Economou, forthcoming). Its foregrounding of visual rules (shape rules drawn in a 2D or 3D modeling system) over symbolic rules (instructions defined in some programming language) provides a robust and disruptive technology for engineers, computer scientists, designers, students, and educators and in general academics and professionals who use drawings and visual models to develop and communicate their ideas.

2 Identity Rules

Seeing in design can be modeled by identity rules—rules that pick-up parts without necessarily doing anything to them (Stiny 1996). The identity rules have identical left-hand sides (LHS) and right-hand sides (RHS) and apply under any given transformation to pick-up parts in a shape seamlessly reorganizing the underlying structure of the shape. The application of a rule follows the general format of a shape production outlined in Stiny (1980). Technically, for shapes A and B, the shape rule $A \rightarrow A$ can apply to a shape B whenever there is a transformation T that makes the shape $T(A)$ part of the shape B. If the shape $T(A)$ is part of the shape B, the rule subtracts the shape $T(A)$ from the shape B and replaces it with the very same shape $T(A)$. The resulting shape B' and the corresponding computation are given in (1).

$$B' = [B - T(A)] + T(A) \tag{1}$$

The computation in (1) captures the notion of a visual query: It restructures a shape in terms of what is the element (shape) that is queried. A series of visual queries are outlined below structured around three conditions: (a) The types of lines that make up a shape, limited here to straight lines, arcs and their combinations; (b) the types of transformations T under which a rule applies, namely isometries and similarities; and (c) the determinacy or indeterminacy of a rule application. Useful overviews of the types of shapes used in shape grammar interpreters are given in Chau et al. (2004) and McKay et al. (2012). A recent discussion on the algebraic language of the group of transformations and its applications in spatial design is given in Pottman et al. (2013). A preliminary discussion on the determinate or indeterminate application of a rule and its relation on the types of intersections between pairs of lines is given in Stiny (2006) and Hong and Economou (forthcoming). A definitive account of the types of shapes based on the possible types of intersections between pairs of lines or arcs is given in Economou and Yu (forthcoming) and a catalogue of n-line shapes for $n \leq 4$ in Economou and Park (forthcoming). Clearly, the examples given below do not exhaust all possible permutations of queries but are selected to give a sense of the expressive power of the search engine of the Shape Machine. For brevity—and to foreground the intuitive aspects of the visual search, these queries are given here in a single shape representation foregrounding the shape that is searched.

2.1 Find Shapes Consisting of Lines Under Isometric Transformations

The first examples of visual queries in Shape Machine are constructed around shapes that consist of straight lines and are searched under isometric transformations in a determinate way. Any shape consisting of straight lines with more or equal than two registration points would do and here the computations shown are based on existing ones in the literature to make the transition from the theory to the application as clear as possible (see, e.g., Stiny 2006). All examples of searches here are based on isometric transformations, that is, transformations that keep the shape and size of a shape but alter its position and/or handedness. Here, this search means that it is confined to congruent instances of the shape that is searched.

The most straightforward shapes consisting of lines to search for are the closed polygons, regular or not. These shapes have typically a name, i.e., triangles, squares, rectangles, rhombi, quadrilaterals, pentagons, hexagons, and so forth, and because of their straightforward geometric structure consisting of well-defined circuits of vertices and edges, they are the very first shapes that have been used to demonstrate the desired ability of a shape grammar interpreter to identify different shapes from the ones used in a two-dimensional model in a visual search (Grasl and Economou 2013). In these cases, the shape grammar interpreter is able to query a specific design in terms of any well-defined polygons. Shape Machine is able to find any closed polygons in a shape even though these polygons may have not been explicitly registered in the database of a CAD system. The example in Fig. 1 showcases one of the possible ways that an initial design consisting of two squares can be queried to reveal possible embedded triangles, squares, pentagons, hexagons, and so forth. The process is straightforward: A pictorial query of a triangle drawn on top of the initial model identifies three more triangles in the model for a total of four. A pictorial query of a square drawn on top of the initial model identifies no more isometric instances of the square in the model. A pictorial query of a pentagon drawn on top of the initial model identifies three additional isometric pentagons in the model for a total of four. The example below shows that a pictorial query of a hexagon in Fig. 1a, in the design

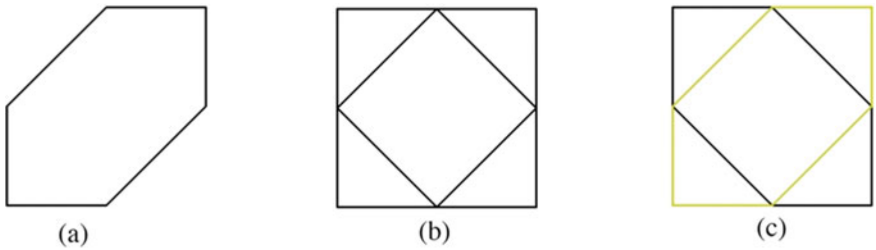


Fig. 1 A query of a closed polyline in Shape Machine under isometric transformations

consisting of two squares in Fig. 1b, identifies two embedded isometric hexagons, one of them shown in Fig. 1c.

The search for polygons can be extended for arrangements of polygons even though these arrangements are not registered in the database of a CAD system. The example in Fig. 2 shows a pictorial query of a spatial relation between two isosceles right triangles reflected along the leading diagonal of an underlying square upon which the triangles are embedded in. Note that this very specific spatial relation can be described in a number of alternative and equivalent ways (Economou and Kotsopoulos 2014). Even more, this spatial relation is just one of several other examples of spatial relations between closed polygons that may be observed in this design, including arrangements of triangles, squares, pentagons, hexagons, and so forth, all sharing a vertex or edge or just floating one to another. In all cases, the Shape Machine can calculate the symmetries of the matches and give the correct number of non-equivalent instances. Here the query shape x consisting of the two right triangles is shown in Fig. 2a, the design that is queried is given in Fig. 2b and one of the two isometric matches of the identity rule is given in Fig. 2c.

Shapes consisting of straight lines need not be well-defined closed polygons or arrangements of polygons. The pictorial query can be extended for continuous lines composed by one or more line segments evoking recognizable symbols such as the letters of the alphabet, for example, A, M, Σ or K, or other symbols from other notational systems. The example in Fig. 3 continues with a query of a three-line shape in the form of the lower-case letter k whose longer edge matches the edge

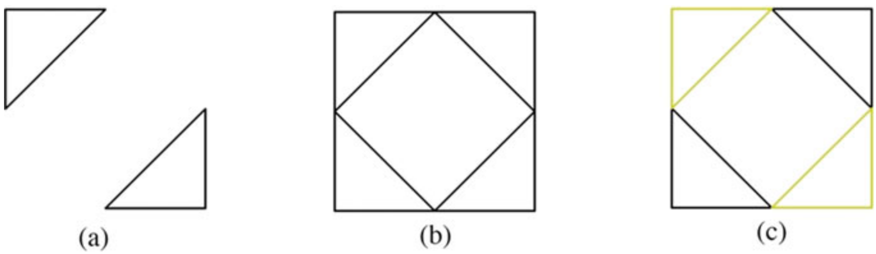


Fig. 2 A query of an arrangement of two polygons in Shape Machine under isometric transformations

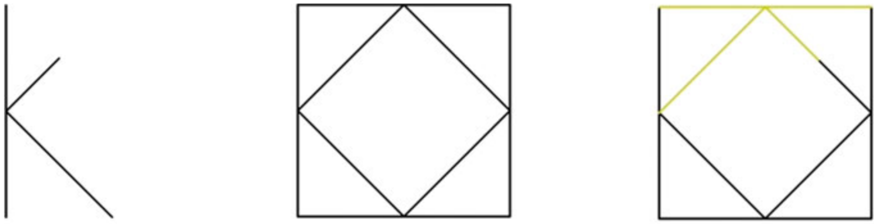


Fig. 3 A query of an arrangement of open polylines in Shape Machine under isometric transformations

of the square, one of its legs matches the edge of the smaller rotated square in the design and the third upright edge has an endpoint (boundary) that is embedded in the edge of the smaller rotated square. Note that the three of the endpoints of the shape *k* as well as its middle 3-valent intersection of its three legs are all registered in the database of the CAD system within the intersections of the lines making up the two underlying squares but the endpoint of the upper leg of the shape *k* in not. The query in Shape Machine yields, respectively, eight instances for each search because the symmetry of the lower-case *k* is equal to 1 and the symmetry of the overall design is 8, and by definition, there are no interactions between the symmetry elements of the symmetry groups of the two shapes—the *k*-shape and the overall design (Stiny 1986). As above, the Shape Machine can calculate the symmetries of the matches and give the correct number of non-equivalent instances. The query *x* of the lower-case *k* is shown in Fig. 3a, the design that is queried in Fig. 3b and one of the eight isometric matches of the identity rule is given in Fig. 3c.

2.2 Find Shapes Under Similarity Transformations

Visual queries need not confine to identical copies. Often designers want to search for similar copies of a shape in smaller or larger versions and in any location and/or possible enantiomorphic or handed versions in a model or series of models. The new unique transformation introduced in the visual query is the scale transformation that varies the size of the shape. Scale transformations combine with the isometric transformations to produce the similarity transformations that keep the shape of a shape but alter its size, position, and/or handedness. The next series of visual queries in Shape Machine are constructed around shapes that consist of straight lines and are searched under similarity transformations in a determinate way. Here, this search means that it is confined to similar instances of the shape that is searched. Note that when these searches are defined for shapes that have some conventional name, say squares, quadrilaterals, and so forth, they retain their semantics; i.e., the search for a particular shape, say a square, is extended for all possible squares in the design.

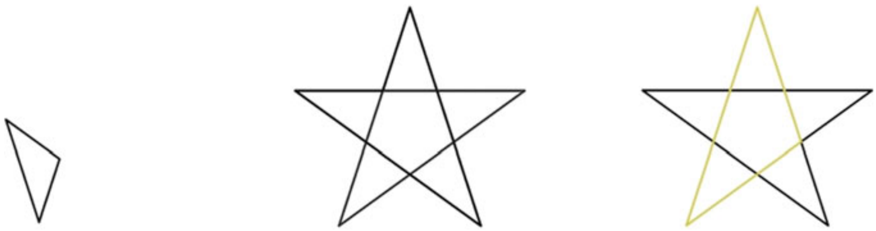


Fig. 4 A query of a polygon in Shape Machine under similarity transformations

The operations of visual queries can be extended for all sorts of shapes consisting of lines. All examples below are structured around the target shape of the 5-star polygon to start from the initial inquiry in Stiny (2006) and continue in a rising complexity from polygons and arrangements of polygons to arrangements of lines.

The most straightforward shapes consisting of lines to search for under similarities are the closed polygons, regular or not. An example of a visual query of a polygon under similarity transformations is given in Fig. 4. Here, the query is an isosceles triangle in with three angles 108°, 36°, and 36° respectively, and the shape that is queried is a regular 5-star polygon. Clearly, several more types of polygons could be searched in this design, for example, convex and concave quadrilaterals, regular pentagons, concave hexagons, concave heptagons, and several more. In all cases, the Shape Machine calculates the exact number of matches without any duplicates and the search here identifies five similar triangles in five different orientations. The query x of the isosceles triangle with angles 108°, 36°, and 36° is shown in Fig. 4a, the 5-star regular polygon design that is queried in Fig. 3b and one of the five similarity matches $T(x)$ of the identity rule is given in Fig. 4c.

The search for similar polygons can be extended for similar arrangements of polygons in any spatial relation without having them registered in the database of a CAD system. An example of a visual query of three polygons in a spatial relation is shown in Fig. 5. The specificity of this spatial relation is quite involved: All three polygons in the spatial relation are isosceles triangles having angles 36°, 72°, and 72°, respectively; two of them pivot around a shared vertex in an angle of 144° so that the endpoint of the short side of one triangle coincides with the endpoint of the long side of the other triangle and both lines forming one continuous line, while a third triangle lies on the middle reflection axis bisecting the angle 144° with its apex at a specific distance l. The discursive description of all this information is indifferent to the Shape Machine that is able to query the shape x in Fig. 5a, in the design in Fig. 5b and give the possible five matches $T(x)$, one of which is shown in Fig. 5c. In all cases, the Shape Machine calculates the exact number of matches without any duplicates: The symmetry of the arrangement of the three triangles is given by the dihedral symmetry group of order 1 and is equal to 2, and the symmetry of the star design is given by the dihedral symmetry group of order 5 and is equal to 10; the

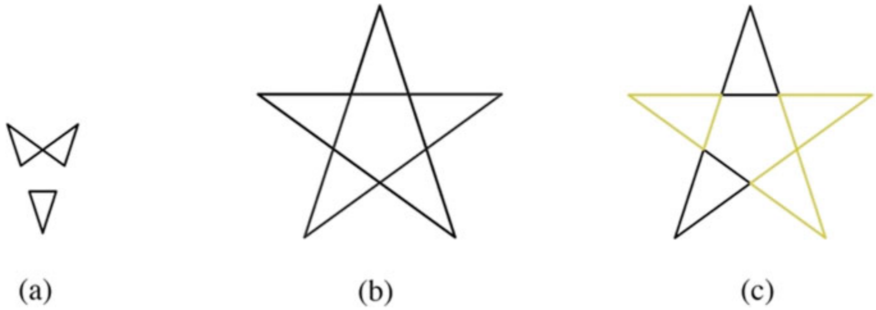


Fig. 5 A query of an arrangement of polygons in Shape Machine under similarity transformations

two shapes share one symmetry element—a mirror refection and the complete non-equivalent matches $T(x)$ are equal to the division of the orders of their symmetry groups equal to 5.

The visual query of shapes consisting of arrangements of lines open can quickly become complex engaging open-ended arrangements of lines that at any desired spatial relation without necessarily specifying some gestalt or conventionalized spatial arrangement. An example of such an arrangement of an L-shape consisting of two lines versus a simple line is shown in Fig. 6. The two lines, the composite one and the straight one, comprise a shape that consists of three lines that require an extensive discursive description to be precise: The three lines lie upon an underlying network of grid-lines forming an isosceles triangle whose inner three angles are 36°, 36°, and 108°, respectively; the projection of the single straight line meets the endpoint of the longest leg of the composite line; and the projection of the short leg of the composite shape meets the straight line in its body. Clearly, these spatial relations between the two lines—the composite and the simple—or the three lines, depending on how one sees it—need more features and parameters to be fully captured, including the lengths of the various parts of the lines in the relations. The structuring of all this information is indifferent though to the Shape Machine that is able to query the shape x in Fig. 6a in the design in Fig. 6b and give the possible ten matches $T(x)$, one of which is shown

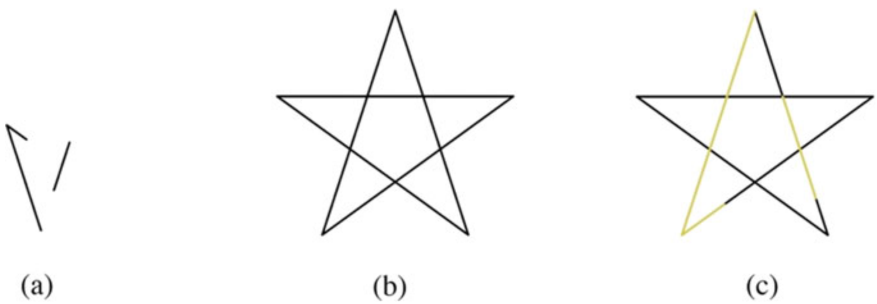


Fig. 6 A query of an arrangement of lines in Shape Machine under similarity transformations

in Fig. 6c. In all cases, the Shape Machine calculates the exact number of matches without any duplicates.

2.3 Find Shapes Consisting of Arcs Under Isometric Transformations

The second family of examples of visual queries in Shape Machine is constructed around shapes that consist of arcs that can be searched under isometric transformations. Any shape consisting of arcs would do, and here as before the computations shown are based on existing ones in the literature to make the transition from the theory to application as clear as possible (see, e.g., Jowers and Earl 2011).

A straightforward class of shapes consisting of arcs to search for under isometries is the closed lens shapes, constructed upon regular or irregular polygons substituting their straight edges with arcs of various lengths. An example of a visual query of a lens or *vesica piscis* shape, a classic figure in Euclid (Fletcher 2004)—and Carlo Scarpa's work for that matter—under isometric transformations is given in Fig. 7. The shape that is queried is a trefoil-like shape consisting of three identical arcs drawn from the vertices of an underlying regular triangle. The radii of the arcs of these shapes are typically equal to the length of the sides of the underlying regular triangle but here are equal to the two-thirds of the edge of the triangle. The symmetry of the *vesica piscis* shape is given by the dihedral symmetry group of order 1 and is equal to 2, and the symmetry of the trefoil design is given by the dihedral symmetry group of order 3 and is equal to 6. The two shapes share one symmetry element—a mirror refection, and the complete non-equivalent matches $T(x)$ are equal to the division of the orders of the two symmetry groups equal to 3. The query x of the *vesica piscis* shape is shown in Fig. 7a, the trefoil design that is queried in Fig. 7b and one of the three isometric matches $T(x)$ of the identity rule is given in Fig. 7c.

Shapes consisting of arcs need not be well-defined closed arrangements of sectors of circles or unions or intersections of sectors of circles. The pictorial query can be extended for continuous arcs composed by one or more arcs. The example in Fig. 8

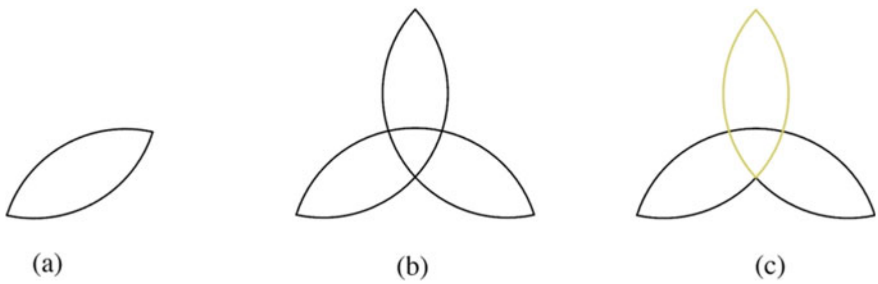


Fig. 7 A query of a closed arc shape in Shape Machine under isometric transformations

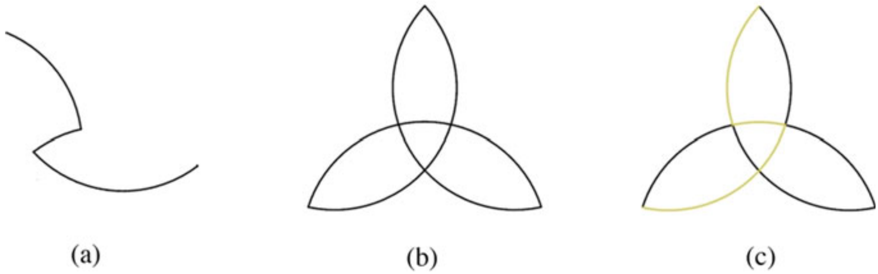


Fig. 8 A query of an arrangement of arcs in Shape Machine under isometric transformations

continues with a query of a three-arc shape in the style of the Mac logo. As before, the description of this shape can be formidable: Here, a chain of three arcs that share two endpoints with alternate convexities and concavities and specific radii and centers upon a regular triangle. Still, the query in Shape Machine is straightforward by simply pointing to the shape that will be queried. The query here yields six results calculating the threefold dihedral symmetry of the three-lens polygon of order 6 by the order of symmetry of the compound arc equal to 1. The query x of the compound arc is shown in Fig. 8a, the trefoil design that is queried in Fig. 8b and one of the six isometric matches $T(x)$ of the identity rule is given in Fig. 8c.

The previous example showed the query of a compound arc whose structural features, including centers of arcs and intersections of arcs, could be potentially retrieved from the database of CAD system. The next example in Fig. 9 showcases the search of a shape consisting of a rotational arrangement of three arcs such that each arc has only one endpoint registered in the database of a shape to be queried. Clearly, this query can be extended for a variety of shapes consisting of three or more arcs or combinations of arcs and straight lines without caring about whether their boundaries, endpoints, and intersections are registered in the database of the CAD system to be searched. The query here yields two results because the threefold dihedral symmetry of the three-lens polygon of order 6 is divided by the order of symmetry of the rotational symmetry of order 3 as long as both shapes share three symmetry elements, namely the rotations of order 0°, 120°, and 240° around their

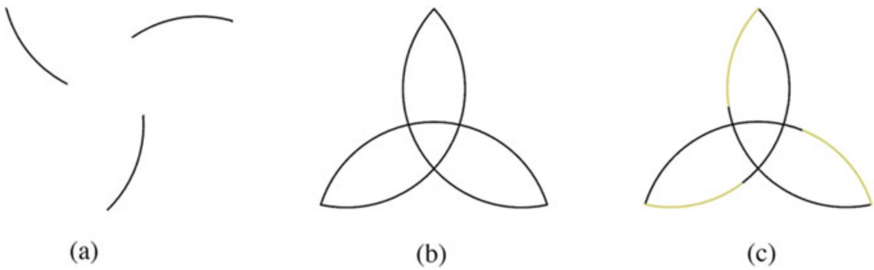


Fig. 9 A query of an arrangement of arcs in Shape Machine under isometric transformations

centers. The query x of the rotational arrangement of the three arcs is shown in Fig. 9a, the trefoil design that is queried in Fig. 9b and one of the two isometric matches $T(x)$ of the identity rule is given in Fig. 9c.

2.4 Find Shapes Consisting of Arcs Under Similarity Transformations

Visual queries of arcs need not confine to identical copies. As before, often designers search for similar copies of a shape in smaller or larger versions and in any location and/or possible enantiomorphic or handed versions in a model or series of models. Interestingly, scale transformations overall appear to affect the queries of shapes in similar ways with the queries of shapes consisting of straight lines but there are significant differences too—that will become apparent in the next section on the determinate and indeterminate application of recognition of identity rules.

The first family of examples of visual queries in Shape Machine of shapes that consist of arcs that can be searched under similarity transformations are again queries of the boundaries of well-defined unions, intersections, or symmetric differences of circles and/or closed polygons. One straightforward example of an intricate Boolean intersection of three circles constructed around an underlying equilateral triangle is shown in Fig. 10. The number of the three similar matches $T(x)$ of this shape is calculated by checking whether the symmetry elements of the query the queried shape intersect, and if yes, by dividing the order of symmetry of the queried shape by the order of symmetry of the query. The query x of the Boolean intersection is shown in Fig. 10a, the trefoil design that is queried in Fig. 10b and one of the three similarity matches $T(x)$ of the identity rule is given in a highlighted form in Fig. 10c.

A second family of examples of visual queries in Shape Machine of shapes that consist of arcs that can be searched under similarity transformations can be constructed by taking any arrangements of arcs irrelevant of whether they specify closed shapes or connected arcs in continuous circuits. Even more, this inquiry can be extended by asking that the boundaries of the arcs (endpoints) are not necessarily

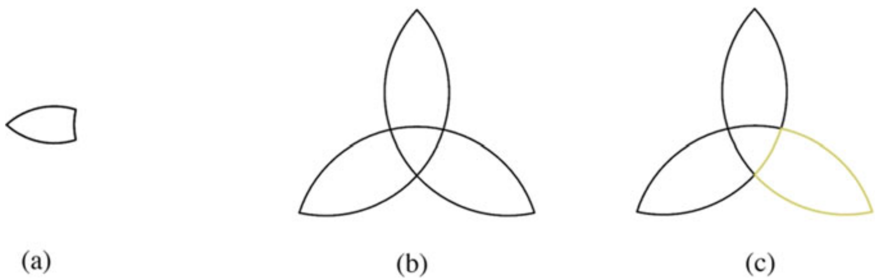


Fig. 10 A query of an arrangement of arcs in Shape Machine under similarity transformations

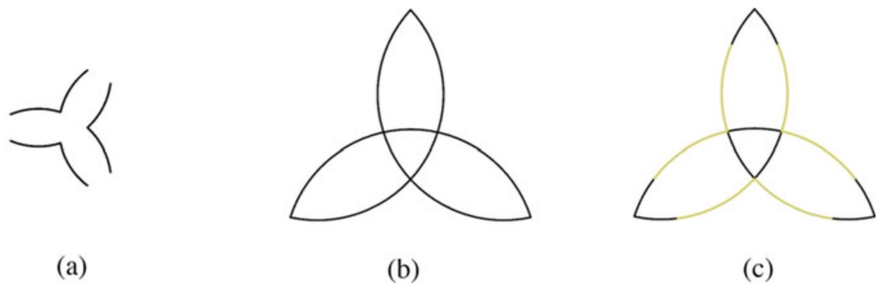


Fig. 11 A query of an arrangement of arcs in Shape Machine under similarity transformations

registered in the database of the CAD system that underlies the specification of a shape that will be searched. An example of a query of such a shape is given in Fig. 11. Here a shape consisting of six arcs is queried with three points defined as the intersections of three pairs of arcs while the other six endpoints are indefinitely defined upon the circumference of the arc. The lengths of the arcs are all the same, and they are identically disposed around a center giving to the shape a dihedral symmetry of order 3; that is, a total symmetry of order 6 equal to the order of symmetry of the trefoil shape that is queried. The query x of the six arcs is shown in Fig. 11a, the trefoil design that is queried in Fig. 11b and the single similarity match $T(x)$ of the identity rule is given in a highlighted form in Fig. 11c.

A last example of a visual query in Shape Machine of shapes that consist of arcs and that can be searched under similarity transformations is considered here by taking any arrangement of arcs irrelevant of specificities of spatial relations between them and to an underlying shape. Here this query shape x is constructed by eliminating parts of a trefoil shape yielding a highly expressive shape that defies a discursive description. Clearly, the symmetry of the emergent shape is trivial and equal to 1 and there are six $T(x)$ matches of this shape in the trefoil shape. The query x of the fragmented shape (arrangements of arcs) is shown in Fig. 12a, the trefoil design that is queried in Fig. 12b and one of the six similarity matches $T(x)$ of the identity rule is given in a highlighted form in Fig. 12c.

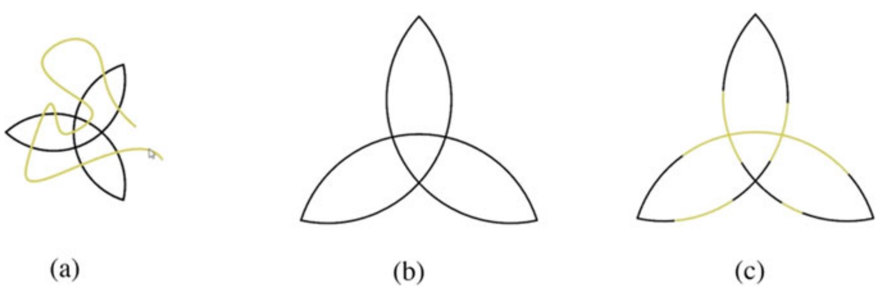


Fig. 12 A query of an arrangement of arcs in Shape Machine under similarity transformations

2.5 Exceptions Noted: Indeterminate Applications of Rules

The results of the queries of the shapes have been so far very successful for a good reason: Most rules apply in a determinate way, and the results can be enumerated and visually inspected. Still, in some cases, the application of the rules is not straightforward and some additional information may be needed to make the exact matches. In these cases, the queries rather than producing a set of instances of shapes $T(x)$ that match the query x, they produce families of sets of instances of shapes $T(x)$. A set of conditions specifying the determinacy or indeterminacy of a rule application is given in Stiny (2006). Any shape rule with a LHS shape consisting of lines in the plane with zero registration points cannot apply in a determinate way under any Euclidean, affine or linear transformation; and any such shape rule with one registration point cannot apply in a determinate way under any similarity, affine or linear transformation—or otherwise, it can apply in a determinate way only under isometric transformations; in either case, additional information is needed to determine the match of the rule.

An example of an indeterminate query can be given in the query of a lower-case k-shape under similarity transformations in Fig. 12. All three lines of the k-shape intersect in a singular point and match the condition above when the lines of the shape intersect in one registration point. The shape k can be matched (found) in eight different kinds of ways in a shape consisting of two squares because of the spatial relations of the elements of the symmetry groups of the two squares and the k-shape. Still, for any of these eight matchings there is an indefinite number of scale transformations that match the k-shape in the shape of the two squares having as a maximal extreme case the matching of the long line of the k-shape with the edge of the large square and as a minimum extreme the arbitrary screen resolution of the k-shape. The query x of the lower-case k-shape is shown in Fig. 13a, the nested square design that is queried in Fig. 13b and one instance $T(x)$ of the eight families of similarity matches with a scalar transformation of 1.5 of the identity rule is given in a highlighted form in Fig. 13c.

These two-step visual queries in Shape Machine need not be confined to the queries of well-structured shapes like the lower-case k-shape under similarity transformations. The queries x can be extended to any shapes consisting of arrangements

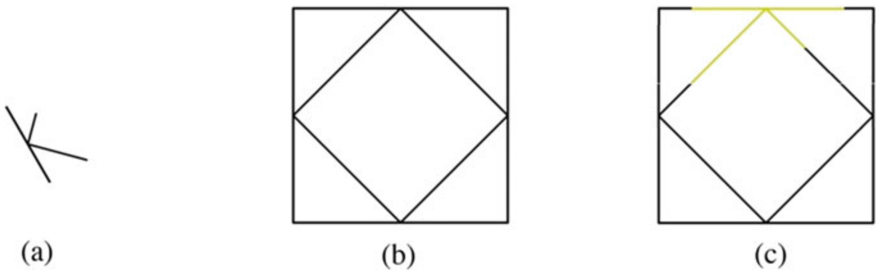


Fig. 13 A two-step query of an arrangement of lines in Shape Machine under similarity transformations

of lines drawn upon construction lines intersecting on a singular registration point. An example of a query of y-shape with an additional floating line emanating from its intersection point is given in Fig. 14. The query x of the y-shape is shown in Fig. 14a, the nested square design that is queried in Fig. 14b and one instance $T(x)$ of the eight families of similarity matches with a scalar transformation of 1.5 of the identity rule is given in a highlighted form in Fig. 14c.

Finally, these two-step visual queries in Shape Machine can be combined with symmetry calculations to provide the answers to queries that entail symmetrical matches and elimination of similar results. An example of such computation is given in Fig. 15. Here, the query is given in the form of a shape consisting of three separate lines, resembling a three-stroke symbol in some symbolic language, with a mirror symmetry of order 2. The query x of the symmetrical shape is shown in Fig. 15a, the nested square design that is queried in Fig. 15b and one instance $T(x)$ of the four families of similarity matches with a scalar transformation of 1.5 of the identity rule is given in a highlighted form in Fig. 15c.

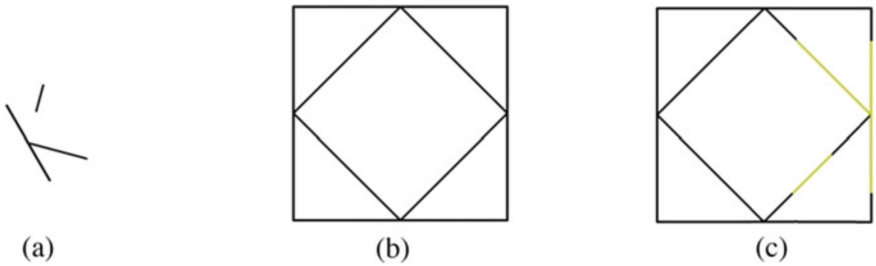


Fig. 14 A two-step query of an arrangement of lines in Shape Machine under similarity transformations

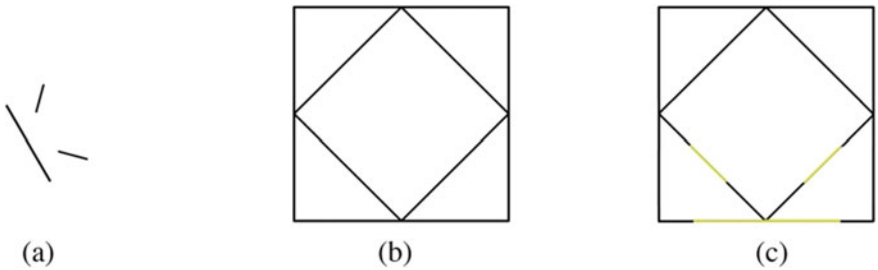


Fig. 15 A two-step query of an arrangement of lines in Shape Machine under similarity transformations

3 Replacement Rules

Doing in design can be modeled by replacement rules—rules that pick-up parts and replacing them with other parts. The replacement rules have different *LHS* shapes and *RHS* shapes and apply under any given transformation to pick-up parts in a shape reorganizing the underlying structure of the shape. The application of a replacement of production rule follows the general format of a shape production outlined in Stiny (1980). Technically, for shapes A and B, the shape rule $A \rightarrow B$ can apply to a shape C whenever there is a transformation T that makes the shape $T(A)$ part of the shape B. If the shape $T(A)$ is part of the shape C, the rule subtracts the shape $T(A)$ away from the shape B and replaces it by the shape $T(B)$. The resulting shape C' and the corresponding computation are given then as:

$$C' = [C - T(A)] + T(B) \tag{2}$$

The replacement rules (doing rules) in (2) are essentially rules in the replacement schema $x \rightarrow y$ (Stiny 2006; Economou and Grasl 2018). As before, the series of replacements outlined below exemplify the three conditions outlined in the seeing rules: the types of shapes involved in the computation (straight lines and arcs), the transformation under which the rules apply, and the types of rules involved in the computation—determinate or indeterminate—and as a consequence, the number of steps required to execute the rule (so far, one-step rules or two-step rules). After all, for a rule $x \rightarrow y$ to apply to a shape A the identity rule $x \rightarrow x$ has to apply under some transformation T and then the resulting shape $T(x)$ can be subtracted from the shape A and replaced by the corresponding transformation of $T(y)$ to produce the shape $A - T(x) + T(y)$.

Significantly, this series of rule applications (visual replacements) may be structured around one extra condition: the recursive definition of the schema y in terms of the schema x, so that the two parts of the rule are related in some way captured in distinct schemata rules (Stiny 2006; Economou and Kotsopoulos 2014). Here, a very brief discussion of these design actions (rule schemata) is given and the focus is given instead in summations of design actions to foreground the visual impact of the rule. Such fractal designs (Mandelbrot 1982) often provide the initial discourse for recursive geometric modeling because of the nature of the ordered repetition of given rules. All examples in this section are extracted from actual computations with the Shape Machine and are not edited in any way: The rules are represented in the classical shape grammar format with the two sides of the production system, *LHS* and *RHS*, the arrow (→) in-between and the two registration marks (+) on either side of the middle arrow (→) to fix the spatial relation between them. The shape rules in Shape Machine can be imported, or they can be defined from scratch.

3.1 Substitute Shapes Consisting of Lines Under Isometry and Similarity Transformations

An initial set of examples of replacement rules in Shape Machine is constructed within the rule schema $x \rightarrow x + t(x)$. These computations are samples of computations of shapes consisting of straight lines and searched under isometry and similarity transformations in a determinate or indeterminate way. The rule in Fig. 16 specifies that a right isosceles triangle should be substituted by the same triangle x plus a half size right isosceles triangle $t(x)$. Note that this specific action can be described equally well in a different of other schemata, for example, $x \rightarrow t_1(x) + t_2(\mathrm{x})$, for t_1 and t_2 similarity transformations of the original right isosceles triangle; or in terms of different operators, for example divisions and boundaries that may divide an initial shape into two different ones or even identical copies of itself as in rep-tile constructions (Gardner 2001). The series of parallel applications of the rule in the initial design comprised of four triangles produces a series of fractal-like designs comprised by $4 + 8 = 12$ triangles, $4 + 8 + 16 = 28$ triangles, $4 + 8 + 16 + 32 = 60$ triangles, and $4 + 8 + 16 + 32 + 64 = 124$ triangles, respectively. The shapes x in the *LHS* and $x + t(x)$ in the *RHS* of the rule are shown in Fig. 16a; one of the four possible matches of the shape $T(x)$ under a similarity transformation T and its corresponding replacement of the shape $T(x + t(x))$ is given in Fig. 16b; and a design after the execution of a series of rules in Fig. 16c.

Both parts of a shape rule, *LHS* and *RHS* are editable in Shape Machine. The example in Fig. 17 demonstrates the design of the *RHS* of a rule in the schema $x \rightarrow y$ or perhaps and more specifically, $x \rightarrow x - prt(x) + y$, for x an isosceles right triangle, $prt(\mathrm{x})$ the part of the hypotenuse of the right isosceles triangle that is erased, and y a new set of lines in some spatial arrangement with the leftover part of the right isosceles triangle. Clearly, this last action can be described in a number of alternative ways (Economou and Kotsopoulos 2014). A single application of the rule in the schema $x \rightarrow \sum T(y)$ in the initial design ends the computation providing a design with a dihedral symmetry of order 8. The shapes x in the *LHS* and y in the *RHS* of the rule are shown in Fig. 17a; one of the four possible matches of the shape $T(x)$ under a similarity transformation T and its corresponding replacement of the shape

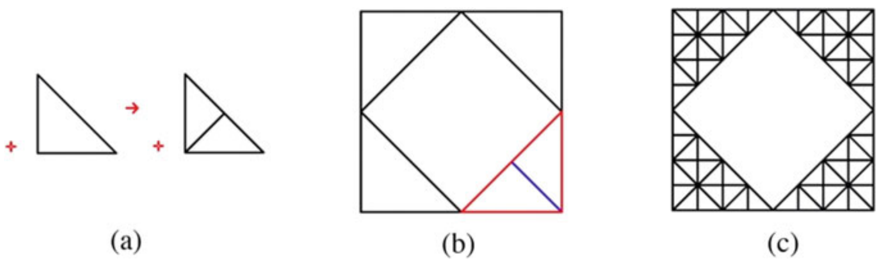


Fig. 16 A construction in Shape Machine using a single shape rule in the schema $x \rightarrow \sum T(x + t(x))$

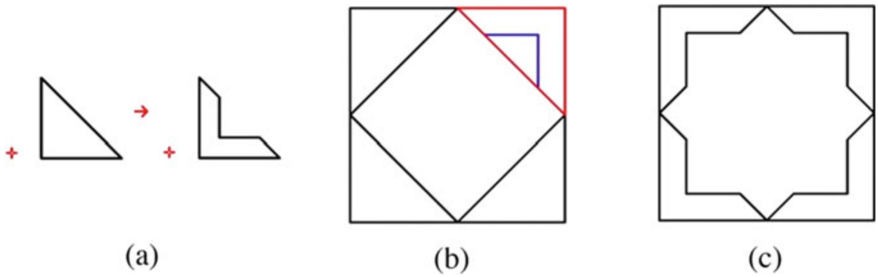


Fig. 17 A construction in Shape Machine using a single shape rule in the schema $x \to \sum T(y)$ in a determinate way

$T(y)$ is given in Fig. 17b; and a final design after the execution of a series of rules in Fig. 17c.

New shape rules can be defined from scratch in the Shape Machine and are editable right away. The construction of a design in Fig. 18 showcases the power of radical redescription Shape Machine can offer in design inquiry. The rule used here to illustrate exemplify this generative redescription is polemically built upon the k-shape, an otherwise rather esoteric illustration of the notion of a radical change of vocabularies in visual composition. Here, the rule specifies that the k-shape in the *LHS* should be replaced by a completely new shape in the *RHS* of the rule that is formed by connecting the endpoints of the k-shape with a set of two disjoint lines. Note that the application of the rule under isometry or similarity transformations provides a sense of surprise to the designer because of the emergent connections of previously disjointed lines to facilitate a reading of the overall design into one continuous folding line. The shapes x in the *LHS* and y in the *RHS* of the rule are shown in Fig. 18a; one of the eight possible matches of the shape $T(x)$ under a similarity transformation T and its corresponding replacement of the shape $T(y)$ is given in Fig. 18b; a final design after the execution of a series of rules is shown in Fig. 18c.

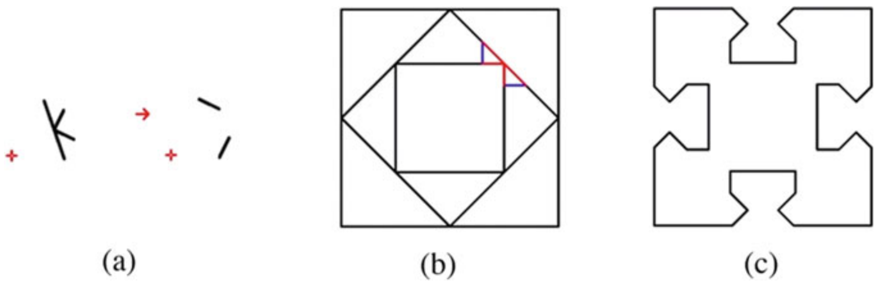
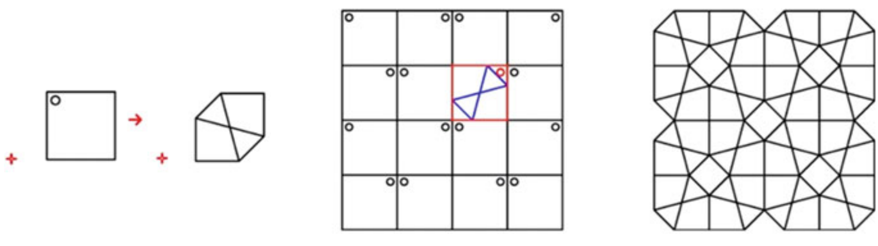


Fig. 18 A construction in Shape Machine using a single shape rule in the schema $x \to \sum T(y)$ in an indeterminate (two-step) way

3.2 Substitute Shapes Consisting of Lines and Arcs Under Isometries and Similarities

A second set of examples of replacement rules in Shape Machine may be constructed around shapes that consist of arcs and straight lines and searched under isometry and similarity transformations in a determinate or indeterminate way. The examples are built upon one of the studies of the original ice-ray grammars (Stiny 1977; Economou and Grasl 2018). The conventions of that analog grammar, including the specification of the initial labeled shape and the labeled rules, are exactly the same with the ones used in this digital grammar. However, the examples are extended here with the additional intent to generalize the design workflow from configuration to ornament demonstrating a seamless process in design. In this first example, the rule is cast in the general schema $x \rightarrow y$ or more specifically $x \rightarrow prt\ x + y$. In this example, the shape x in the *LHS* is a square with a circular label on the top left corner and the shape y in the *RHS* is an elongated hexagon with an internal set of diagonals to create two triangles and two quadrilaterals (and many more shapes as well)—all with a dihedral symmetry of order 4. The shape to be searched in terms of this rule is a 4 × 4 square grid with alternating labels ensuring a diagonal reflectional pattern. The shapes x in the *LHS* and y in the *RHS* of the rule are shown in Fig. 19a; one of the sixteen possible matches of the shape $T(x)$ under a similarity transformation T and its corresponding replacement of the shape $T(y)$ is given in Fig. 19b; a final design after the execution of a series of rules is shown in Fig. 19c.

A simple reworking of the same schema rule with a substitution of a shape made of straight lines in the *LHS* with a shape made up for arcs in the *RHS* can produce a radical redescription in a design. The substitution of the square shape x with a circular label on its top left corner in the *LHS* of a shape rule with a *vesica piscis* shape composed by two arcs with centers the two diagonal vertices of the square and radii the edges of the square in the *RHS* produces a rule that when it is applied in all sixteen parts of a 4 × 4 alternating square grid as described above, makes a completely new design that features the same underlying symmetry but with very different expressive qualities. Significantly, none of the underlying straight lies survive the parallel



(a) (b) (c)

Fig. 19 A construction in Shape Machine using a single shape rule in the schema $x \rightarrow \sum T(y)$ in a determinate way

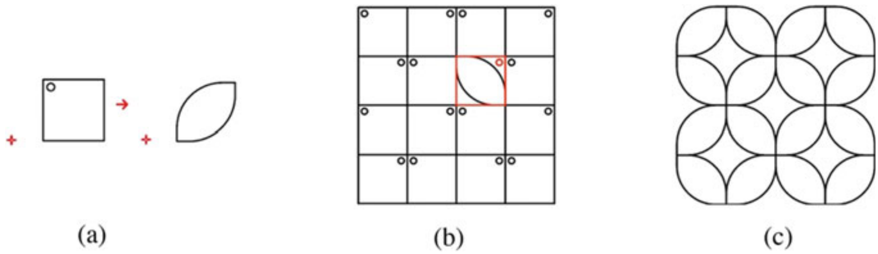


Fig. 20 A construction in Shape Machine using a single shape rule in the schema $x \rightarrow \sum T(y)$ in a determinate way

replacement of all the rule applications and the new design acquires a very different feel and look. The shapes x in the *LHS* and y in the *RHS* of the rule are shown in Fig. 20a; one of the 16 possible matches of the shape $T(x)$ under a similarity transformation T and its corresponding replacement of the shape $T(y)$ is given in Fig. 20b; a final design after the execution of a series of rules is shown in Fig. 20c.

These replacement rules need not confine to well-structured or conventionalized geometries as in the 4×4 alternating square grid discussed above. Shapes made of arcs can be transformed to straight lines on the spot and the other way around too if a shape rule that does so apply to them. Both translations are typical with the schema rule $x \rightarrow y$ and/or $x \rightarrow \sum T(y)$ if the rules are applied in a fractal way. An example illustrating this translation between geometries is given in Fig. 21. In this example, the part of an arc of a circle in the *LHS* of a rule is substituted by a three-line bracketed shape. When this rule applies in the design generated in the previous example, it generates a radically different lattice that keeps the topology of the previous pattern. The shapes x in the *LHS* and y in the *RHS* of the rule are shown in Fig. 21a; one of the 32 possible matches of the shape $T(x)$ under a similarity transformation T and its corresponding replacement of the shape $T(y)$ is given in Fig. 21b; a final design after the execution of all possible rule applications is shown in Fig. 21c.

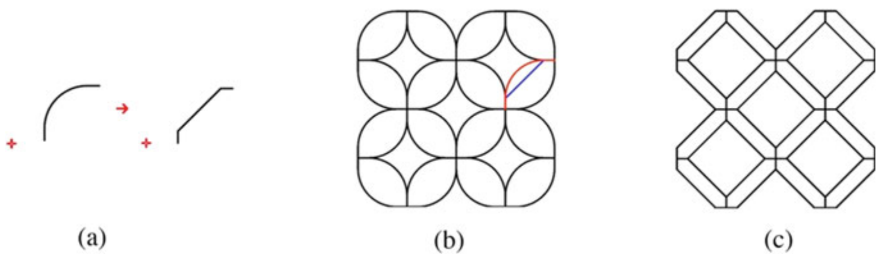


Fig. 21 A construction in Shape Machine using a single shape rule in the schema $x \rightarrow \sum T(y)$ in a determinate way

4 Applications

A very different discussion of rule replacements—and design actions—may be considered when the focus is directed instead in the calculation of specific examples in design domains such as ornamental design, product design, CAD drafting, architectural design, mechanical design, electrical design, and so forth. Few selected examples are given below to showcase the versatility and expressiveness of the Shape Machine in distinct design flows in an interactive mode as well as in a fully automated mode suggesting the promise of a complete visual programming language and/or a new paradigm in computing machinery.

4.1 Product Design: Celtic Knots

A product design application to generate a Celtic knot weave is presented in the form of a shape grammar implemented in the Shape Machine. Celtic knots and their spirals, step patterns, and key patterns are some of the most enduring stylized graphical representations of knots in ornamental art (Bain 1975). A shape grammar for Celtic knots has been proposed by Jowers and Earl (2011) and provides the blueprint for its implementation in Shape Machine. The grammar is modeled in three stages: the first stage fixes the central horizontal and vertical growth of the pattern. The second takes care of the corner conditions retaining the motion of the alternating loop throughout the pattern. The last stage modifies the intersections of the loops to disentangle the knot and begin to show how spirals, step patterns, and key patterns may be treated in the composition. The shape rules consist of shapes composed of arcs and straight lines and apply to the working model (design) under similarity transformations. Interestingly, the rules appear that work within the schema $x \to x + t(x)$, for x a loop and $x + t(x)$ a double loop, but the details are rendered somewhat differently to account for the alternating motion of the loop.

The first stage of Celtic knot grammar captures the underlying central horizontal and vertical growth of the pattern and consists of two shape rules that fix the horizontal and vertical growth of the pattern. Clearly, the rules can be applied several times to generate large cruciform configurations of weaves. Both shape rules have been designed directly in Shape Machine using an initial shape of the two loops as a construction shape, and they apply to the evolving design under similarity transformations. The open-loop shape x in the *LHS* and the open double-loop shape y in the *RHS* of the rule are shown in Fig. 22a; one of the two possible matches of the shape $T(x)$ under a similarity transformation T and its corresponding replacement of the shape $T(y)$ is given in Fig. 22b; a final design after the execution of two possible rule applications is shown in Fig. 22c.

The second stage of the Celtic knot grammar captures the field growth of the pattern from its central horizontal and vertical axes to its boundary extents. Two additional rules are introduced to capture the growth of the pattern along its minor

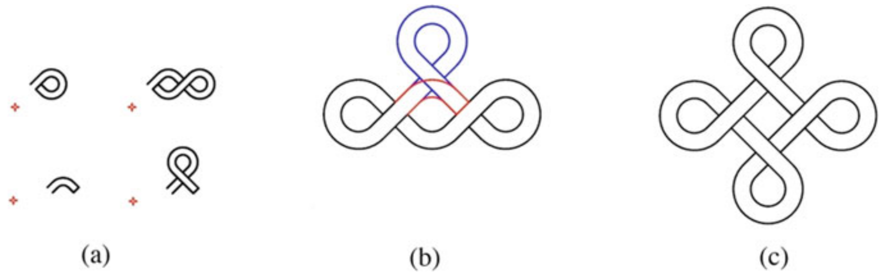


Fig. 22 Celtic knot grammar: Configuration stage

horizontal and vertical axes and resolve the corner conditions keeping throughout the alternating loops of the pattern. Note that the derivation of the pattern could terminate at the will of the designer to give a rotational symmetry to the overall design—in addition to the rotational symmetry of the inner pattern—or as above, to complete the boundaries and produce a composite symmetry: a square symmetry for the underlying configuration, a vertical symmetry for the loops and a rotational symmetry for the joints. As before, the shape rules apply to shapes composed of arcs and straight lines under similarity transformations. The two new rules are shown in Fig. 23a, a matching of one of the two rules in the evolving design is shown in Fig. 23b, and a design after the execution of four rules is shown in Fig. 23c.

The last stage of the Celtic knot grammar modifies the intersections of the loops to produce shapes that begin to capture the spirals, step patterns, and key patterns that are characteristic in the composition. Two additional rules are introduced here to show this possible editing of the grammar and the disentanglement of the knot. Both rules are symmetrical and reverse copies of each other each undoing what the other is doing, i.e., the one untying the knot and the second tying it back reversing its action. The lengths of the strings of the knots (the two pairs of parallel lines denoting the ends of the two strands of the knot) at the *LHS* of the rules are specified by an area selection action on the model itself to capture the area of intervention of the designer. These subshapes, once captured in the design model, are literally carried on the *LHS* of the rule space to be acted upon. The calculation, as before in this

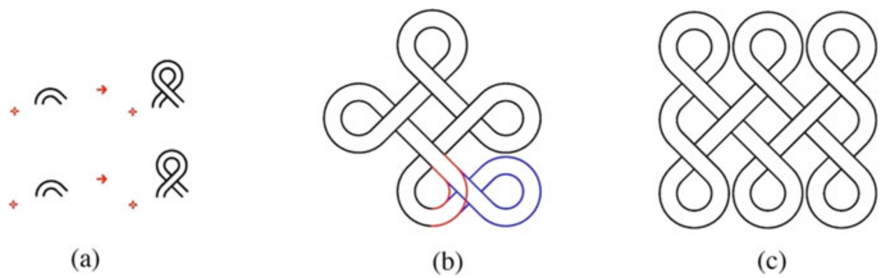


Fig. 23 Celtic knot grammar: Fabric stage

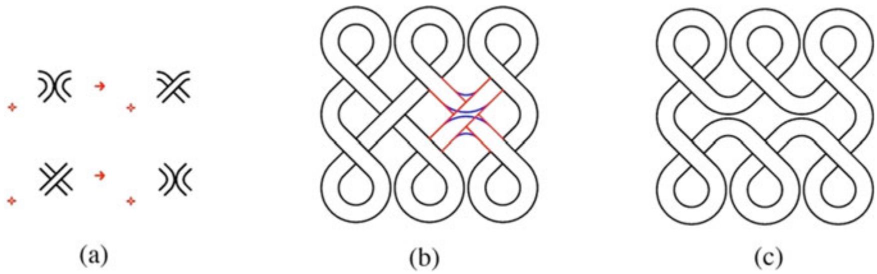


Fig. 24 A Celtic knot grammar: Modifying stage

series, is limited in matchings under similarity transformations. The two new rules are shown in Fig. 24a, a matching of one of the two rules in the evolving design is shown in Fig. 24b, and a design after the double execution of the untying rule is shown in Fig. 24c.

4.2 Mechanical Design: The Gear Grammar

A mechanical design application to generate a gear is presented in the form of a shape grammar implemented in the Shape Machine. The grammar consists of four rules: The first shape rule draws an initial circle defining the outer radius of the gear. The second shape rule offsets a 24-gon to define the inner radius of the gear. The third shape rule subdivides the outer radius to specify twelve wedged parts. Finally, the fourth shape rule specifies the design of a tooth at the outer boundary of the wedge. The mechanical gear is generated by the serial application of shape rules 1–3 one at a time, and the parallel application of shape rule 4 to all twelve parts of the design. Significantly, these rules are grouped and executed in the *DrawScript* mode of Shape Machine, an automated setting that executes rules under specifications given by the designer. The pictorial template allows for the visual specification of the shape rules along with numerical specification for the number of loops and the types of transformations under which the rules apply. When the visual program runs, the rules are applied sequentially to: (1) draw an initial circle defining the outer radius of the gear; (2) offset a second circle to define the inner radius of the gear; (3) subdivide the outer radius to specify twelve parts; and (4) generate teeth at each of the twelve parts. The fourth rule of the gear grammar is shown in the template of the *DrawScript* mode of the Shape Machine in Fig. 25a. One of the matchings of the fourth rule of the grammar that substitutes one of the twelve sectors of the circle with the pentagonal polyline that outlines the tooth of the gear is shown in Fig. 25b. The complete automated generation of the gear in Shape Machine is given in Fig. 25c.

At any time during the design process, the designer can intervene in the *DrawScript* mode to change a shape rule and create a variation of the design. Here, the last shape rule that specifies the end of the gear is slightly edited to substitute its straight end

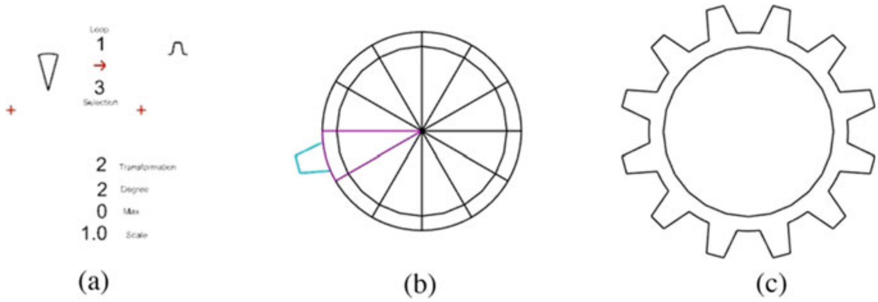


Fig. 25 An automated specification of a mechanical gear in the *DrawScript* mode of the Shape Machine

with an arc end to specify a second design variation. The editing of the profile takes place in real time with the same visual modeling tools of the application. Once the editing of the individual rules is done, the complete set of rules is selected and the program reruns to produce the final design. The edited rule that changes the profile of the tooth is shown in Fig. 26a. One of the twelve matches of the last rule of the gear grammar is shown in Fig. 26b. The new generated variation of the gear is given in Fig. 26c.

The variation on the design of the gear can be further expressed visually by adding new rules, subtracting rules or modifying rules. Here, a structural variation is given by editing the last rule of the template and the substitution of the pie-shaped part of the circle with a polyline composed of seven lines to produce the profile of two teeth. Note that these changes are all done visually, they do not require any programming skills and provide a parametric design agency to the author or user of the grammar. This edited rule is shown in Fig. 27a. One of the twelve matches of the last rule of the gear grammar is shown in Fig. 27b. The new generated variation of the gear with the double order of symmetry, a dihedral symmetry of order 24, is given in Fig. 27c.

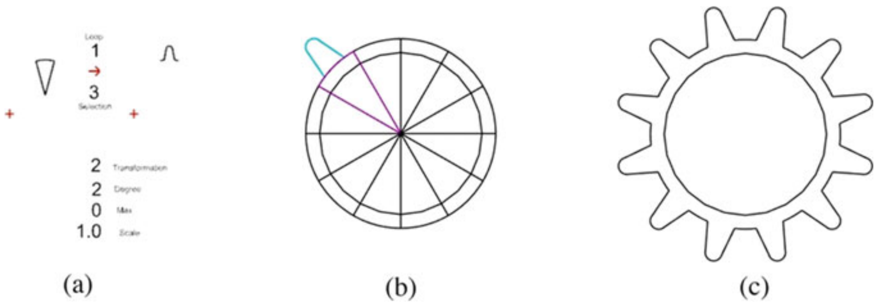


Fig. 26 Editing of the profile of the teeth of a gear in the *DrawScript* mode of the Shape Machine

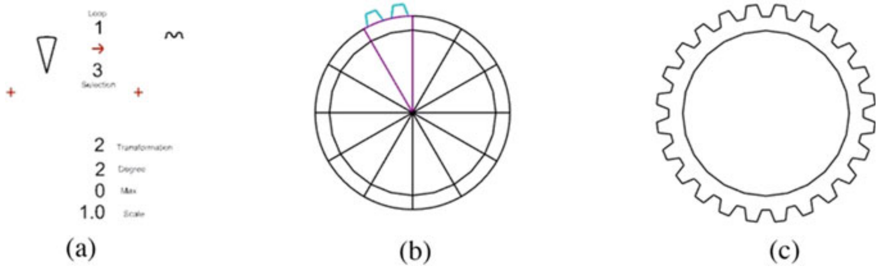


Fig. 27 Editing of the number of teeth of a gear in the *DrawScript* mode of the Shape Machine

4.3 Architectural Design: Portm-Ino

An architectural design application to generate the house plan of a unique private residence is presented in the form of a shape grammar implemented in the Shape Machine. The house, *Entelechy I*, designed by the architect John Portman back in the sixties, holds a unique position in the oeuvre of his office because of the belief of the architect that it encapsulated in some hybrid form the architectural elements of his language that were popularized in his immensely successful commercial architecture starting in the seventies and continued to this day (Portman and Barnett 1976). The generation of the plan of the house is presented in the form of a shape grammar in three stages that roughly correspond to three stages of design and the consolidation of an abstract idea into an architectural plan (Ligler and Economou 2019). The grammar is called Portm-Ino grammar to suggest how the house in Portman's conception is a systematic, residential configuration along the lines of Le Corbusier's Dom-Ino framework.

The first stage of the shape grammar, *Framework*, establishes a basic tartan grid that enables the underlying division of the house in major and minor spaces and the delineation of compositional enfilades to orchestrate the points of centrality within the house. A variety of $n \times m$ grids may be generated in the style and all can be used to be encoded with the layered tartan grid. One of the rules in this stage substituting a module of the underlying grid to a stellated aggregation of five cells with a major cell in the middle and four cells in the diagonals—the basic formation of the major–minor space—is given in Fig. 28a. One of the nine matches of this rule is shown in Fig. 28b. The new generated variation of the tartan grid plan with the nine major spaces is given in Fig. 28c.

The second stage of the grammar, *Configuration*, establishes the major organizational scaffold of the house with its functional division in public and private zones, entries, and its visual articulation in terms of the major and minor cells in terms in the form of an underlying grid punctuated by circular spaces. One of the rules in this stage substituting the boundary of a minor cell with a circular room—or in Portman's words, a hollow or exploded column—an architectural scaffold for lightwells, stairwells, closets, studies, libraries, bathrooms, and micro-galleries, is given in Fig. 29a. One of the sixteen matches of this rule is shown in Fig. 29b. The new generated

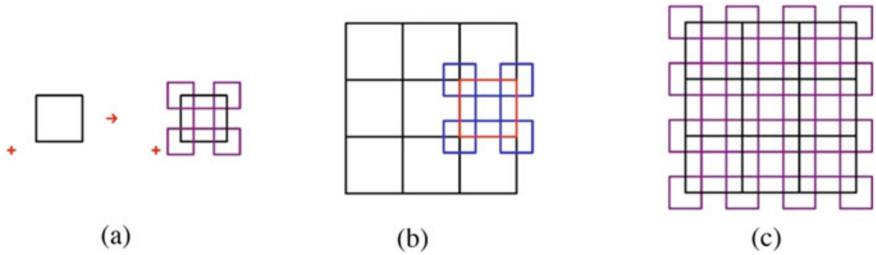


Fig. 28 Portm-Ino: Framework stage

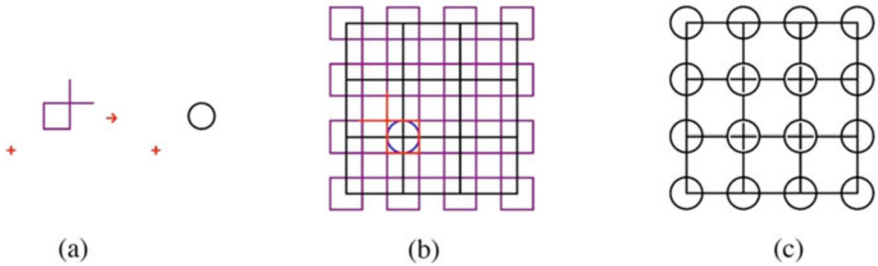


Fig. 29 Portm-Ino: Configuration stage

variation of the gridiron plan with its characteristic arrangement of an underlying grid and circular spaces is given in Fig. 29c.

The third stage of the grammar, *Architectonics*, completes the generation of the plan. This stage consists of shape rules that encode the conventions characterizing the articulation of an architectural plan and includes rules for the delineation of double-height living spaces (denoted by an x in the plans), walls, openings, staircases, bridges, and so forth, to fully disambiguate the design in terms of room of public rooms, living rooms, bedrooms, stairwells, closets, studies, libraries, half bathrooms, and micro-galleries. One of the rules in this stage showing the substitution of a hollowed column with a fully articulated staircase is shown in Fig. 30a. One of the three possible matches of the rule is shown in Fig. 30b. The complete automated production of a possible 3×3 gridiron plan in the language is shown in Fig. 30c.

5 Discussion

A brief survey of the expressive power of Shape Machine has been presented. The work was presented in two parts: A brief presentation on the implementation of the elusive shape computations that have been envisioned by the academic community since the seventies—and still used as the testbed for the tasks a shape grammar interpreter should be able to accomplish (Gips 1999; Stiny 2006). And a second

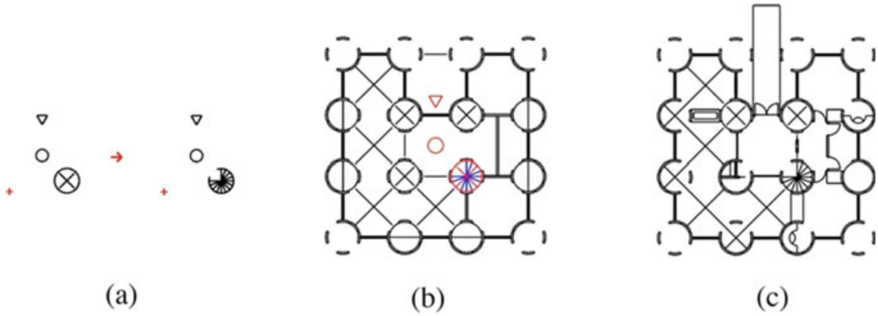


Fig. 30 Portm-Ino: Architectonics stage

part on the exploration of applications in various design domains, namely product modeling, mechanical and architectural design, to suggest possibilities for new design workflows. Clearly, these applications were only given as samples of transformative possibilities rather than as actual design applications. A more generous plateau of possible application domains including, and not limited to, origami structures, archaeology reconstructions, protein folding, virus recognition and more, is currently under development.

The future of the Shape Machine appears bright indeed: It has succeeded to provide a coherent framework to do the elusive computations that no other technology has been able to do so far; and it has managed to do so convincingly in run time promising the possibility of a real application that can indeed revolutionize design thinking and practice. The future steps are already within reach: three-dimensional geometry; *b*-rep representation; solid modeling; larger datasets of geometry primitives; implementation of general transformations for both identity and replacement rules. Some of these questions pertain to practical matters deploying the existing algorithms behind the Shape Machine into new tasks, scales, and dimensions. Others point to the solution of still new theoretical aspects of rule recognition and replacement for all sort of shapes under different transformations.

Current development in Shape Machine aims at the completion of shape rule recognition and implementation algorithms for lines and arcs under the complete Erlangen program, that is, under affine and projective transformations too. The unification is very promising: Existing modules of the Shape Machine, not presented here, suggest intuitive spatial searches in the *Napkin Draw* mode, whereas designers are able to query shapes under compress, shear, and perspectivity transformations inquiring and performing on shape type definitions according to essential characteristics defined on spot (Mitchell 1992). Queries including quick sketches of polygons, and/or arrangements of lines and arcs rather than precise constructions as outlined in this paper are within reach and the line and arc shapes that are used as queries or targets are expanded to include ellipses and conics at large. The work so far promises to completely revolutionize the current work on the Shape Machine and its rule implementations in the Euclidean space because shape computations under affine and projective transformations produce shapes that do not share any more the same conventions with the ones that started the computation; after all, a square is a square no matter if it is rotated, reflected, diminished, or enlarged. But drawing a quadrilateral in the Identity mode (seeing) and having the Shape Machine finding, say, squares or rhombi under affine or projective transformations, respectively, requires some more reworking on behalf of the designer who works more with language (symbolic descriptions) rather than shapes.

But more than those fronts, the work on the Shape Machine front so far has opened two completely new trajectories not fully appreciated in the setting of the inquiry. The one is the work on visual scripting and the *DrawScript* mode. The ability to learn to script through drawing, the interface with loops, what-ifs, control structures and all, suggest an unprecedented environment for teaching computing and interfacing with computers. Sorting operations are important algorithmic constructs in programming, and they can be done in a straightforward manner in Shape Machine: Instead of writing code to sort binary digits, Shape Machine allows users to visually program this logic by considering the symbols 0 and 1 as shapes drawn in the shape rules: 10 → 01 or 01 → 10. The first rule when applied multiple times over a string of shapes made up of 0 s and 1 produces the so-called big-endian order consisting of all 0 s to the left and 1 s to the right. The second rule when applied multiple times over a string of shapes made up of 0 s and 1 s produces the so-called little-endian order consisting of all 1 s to the left and all 0 s to the right. But more than that, these shape rules can be used with additional shape rules to demonstrate that numerical calculation can be done visually! Figure 31 shows an example of such a visual calculation: In this example, a single-digit adder is implemented using 29 shape rules accomplishing three different main operations: (a) translating the numerals (1, 2, 3, …, 9) to numbers of 1 s (1, 11, 111, …, 111111111); (b) moving all the 1 s together; and (c) translating numbers of 1 s (1, 11, 111, …, 111111111) to numbers (1, 2, 3, …, 9, 10). The concept of addition, moving all the 1 s together, is commonly used in a digital environment. Shape Machine can process this operation visually with the simple shape rule discussed above in the visual sorting to achieve an adder. Two of the 29 rules are shown in Fig. 31a. The initial shape denoting an addition of two numbers (7 + 9) is shown in Fig. 31b. Two of the several visual substitutions of the operation and the final result are shown in Fig. 31c. Not that in all these computations, the symbol 7 is treated a shape consisting of two lines, the symbol + by two lines and the symbol 9 by a circle and a line.

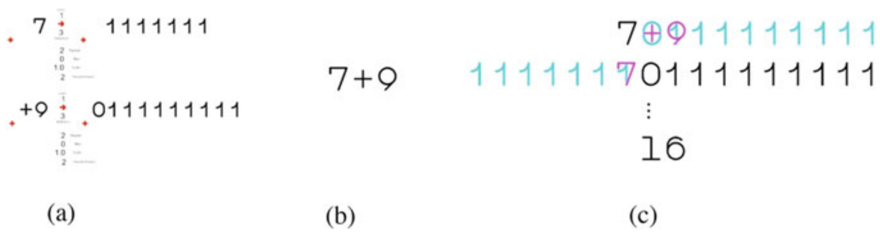


Fig. 31 A visual calculator: Toward a new Turing Machine

This example is a demonstration that implementing a computer with a shape grammar is very possible in theory. And with that, a much deeper and farfetched question: Can we implement a visual computer and how different would it be? Wittgenstein came close to an arithmetic of shape in his remarks to the Foundation of Mathematics but in the end he discarded the possibility as an idle speculation (1956). We strongly believe that the insight this technology can bring will be invaluable to the things we do, but even more, to things we have not dreamed of yet.

Acknowledgements This work has been partially supported with grants by the NSF/I-Corps Site and the Digital Building Lab (DBL) at Georgia Institute of Technology. We would like to thank Pr. Josephine Yu and her students Cvetelina Hill, Nicholas Liao and May Cai at the School of Mathematics at Georgia Institute of Technology, for their valuable insights and advice on the project.

References

Bain, G. (1975). *Celtic art: The method of construction*. London: Lewis Reprints Ltd.

Chau, H. H., et al. (2004). Evaluation of a 3D shape grammar implementation. In J. S. Gero (Ed.), *Design computing and cognition'04* (pp. 357–376). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Earl, C. F. (1997). Shape boundaries. *Environment and Planning B: Planning and Design, 24,* 669–687.

Economou, A., & Kotsopoulos, S. (2014). From shape rules to rule schemata and back. In J. S. Gero & S. Hanna (Eds.), *Design computing and cognition DCC'14* (pp. 419–438). Springer.

Economou, A., & Grasl, T. (2018). Paperless Grammars. In J.-H. Lee (Ed.), *Computational studies on cultural variation and heredity* (pp. 139–160). Springer, Singapore: KAIST Research Series.

Economou, A, & Park, J. (Forthcoming). A pictorial catalogue of *n*-line shapes for *n.*

Economou, A., & Yu, J. (Forthcoming). Shape signature.

Eloy, S., Pauwels, P., & Economou, A. (eds). (2018). Advances in implemented shape grammars: solutions and applications. *AIEDAM Special Issue: Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 32*(2).

Fletcher, R. (2004). Musings on the Vesica Piscis. *Nexus Network Journal, 6*(2), 95–110.

Grasl, T., & Economou, A. (2013). From topologies to shapes: Parametric shape grammars implemented by graphs. *Environment and Planning B: Planning and Design, 40*(5), 905–922.

Gardner, M. (2001). *"Rep-tiles", The colossal book of mathematics: Classic puzzles, paradoxes, and problems* (pp. 46–58). New York: W. W. Norton.

Gips, J. (1999). Computer implementation of shape grammars. *Workshop on Shape Computation*, MIT.

Hong, T. K., & Economou, A. (Forthcoming). Shape machine.

Hong, T. K., & Economou A. (Forthcoming). Five criteria for shape grammars interpreters.

Jowers, J., & Earl, C. (2011). Implementation of curved shape grammars. *Environment and Planning B: Planning and Design, 38*, 616–635.

Knight, T. (1994*). Transformations in design: A formal approach to stylistic change and innovation in the visual arts*. Cambridge University Press.

Krishnamurti, R. (1981). The construction of shapes. *Environment and Planning B: Planning and Design, 8,* 5–40.

Ligler, H., & Economou, A. (2019). From drawing shapes to scripting shapes: Architectural theory mediated by shape machine. In *Conference Proceedings of the Symposium on Simulation for Architecture and Urban Design (SimAUD)* (pp. 278–286).

McKay, A., Chase, S., Shea, K., & Chau, H. H. (2012). Spatial grammar implementation: From theory to useable software. *AI EDAM-Artificial Intelligence Engineering Design Analysis and Manufacturing, 26*(2), 143–159.

Mandelbrot, B. (1982). *The fractal geometry of nature*. Times Books.

Mitchell, W. (1992). *The logic of architecture: Design, computation, and cognition*. Cambridge: MIT Press.

Portman, J., & Barnett, J. (1976). *The architect as developer*. New York, NY: McGraw-Hill.

Pottmann, H., Asperl, A., Hofer, M., & Kilian, A. (2013). *Architectural geometry*. Bentley Institute Press.

Stiny, G. (1977). Ice-ray: A note on the generation of Chinese lattice designs. *Environment and Planning B: Planning and Design, 4,* 89–98.

Stiny, G. (1980). Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design, 7,* 343–351.

Stiny, G. (1986). A new line on drafting systems. *Design Computing, 1,* 5–19.

Stiny, G. (1996). Useless rules. *Environment and Planning B: Planning and Design, 23,* 235–237.

Stiny, G. (2006). *Shape: Talking about seeing and doing*. Cambridge, MA, USA: MIT Press.

Wittgenstein, L. (1956). *Remarks on the foundations of mathematics*. Oxford: Basil Blackwell.

Shape Grammars as the Decoder of Cultural DNA of Archaeological Artifacts

Myrsini Mamoli

1 Introduction

In general, archaeologists think in the opposite direction of designers. Designers generate designs within a particular style or visual language. Archaeologists work with a set of artefacts, from which they try to extract the common design principles and define a style, whether the medium is pottery, sculpture, painting, or architecture. In a way, we can say that they are trying to deduct the DNA of a culture through the reconstruction of a series of fragments and the comparison of parallels.

In this process of reconstruction of the original state of an artefact or any scale, there is a high degree of uncertainty and ambiguity and archaeologists take several steps of interpretation: first, they document the archaeological record, as descriptively as possible. They clarify the stratigraphy and document the different layers of data with different building phases, following the principle that later phases might include part or all of the earlier phases, but earlier phases cannot include part of the later phases. The quality and the degree of preservation of the archaeological record can differ dramatically from case to case. Then, they analyse the available data and try to fill in the gaps of the missing evidence relying on analogy to parallel artefacts to supply ideas they are missing from the artefact under consideration. They propose the initial state of an artefact based upon its type, structure, technologies, materials, ornaments, and scale in comparison with other similar objects.

Often the available evidence is enough to lead the experts to a certain direction of interpretation and reconstruction, but not enough to narrow down the possibilities to just one. Experts disagree about the interpretation of available evidence and the resulting conjectures about an artefact's original state. There are many ways to complete a picture, and the various ways that architects or archaeologists complete

M. Mamoli (✉)
School of Architecture, College of Design, Georgia Institute of Technology, Atlanta, USA
e-mail: myrsini@gatech.edu

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_7

the picture of a representation of an artefact or building in some initial state can differ dramatically and can produce different results. Therefore, a substantial gap remains between the representation of the fragmentary existing object, produced through fieldwork, and the proposed representation of a proposed initial state.

Iteration and multiple variant reconstructions become key to the communication of the results of archaeological research, in an area that uncertainty will always be dominant. Wescoat in her analysis and reconstruction of the Ionic Porch and the Nike Precinct at the Sanctuary of the Greater Gods in Samothrace chose to present multiple iterations that offer different realizations of the buildings, rather than present the most possible among them, indicating areas of uncertainty with various degrees of opacity (Wescoat 2019).

Another example is the Ancient Roman Library of Nysa. A series of archaeological excavations since the beginning of the twentieth century have resulted in the identification of a building in the west side of the city of Nysa as the library. Identification of this building with the ancient library is revealed through its striking similarity with the contemporary libraries at Ephesus and Sagalassos, both built in the second century CE and geographically relatively close to Nysa. These similarities include the podium, the niches, and auxiliary support spaces, all significant characteristics of Hellenistic and Roman libraries. Despite this evidence, over the history of the excavations, Diest, Hoepfner, and Idil have proposed three distinct theses about the possible designs of its initial state (Fig. 1) and still there is no consensus that the building was a library and about its initial form.

These interpretations clearly offer different realizations of the building. Moreover, as suggested by the corpus of the identified libraries in the Roman world, other variations are possible too. But, if these proposed realizations (and more) are all possible, how can we begin to reason about these possible interpretations? Upon which premises is one solution better than others? What are the steps, if any, for the modelling of such reconstructions? Is it possible to have such assertions formalized and agreed upon?

Questions like those have been increasingly taken on by computational and formal methods in archaeological discourse. The introduction of formal methods in archaeological theory started in the second half of the twentieth century as part of the

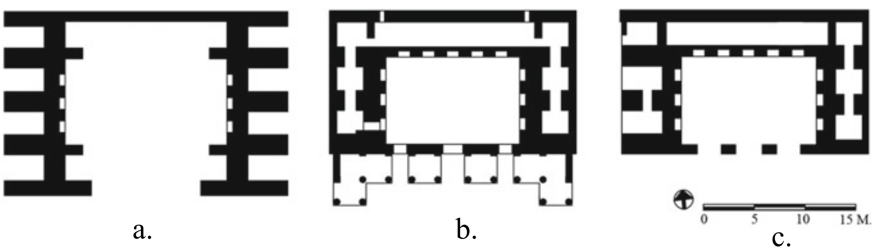


Fig. 1 Alternative realizations of the plan of the Library of Nysa, redrawn by the author after Diest, Hoepfner, and Idil (Mamoli 2014, Fig. 4.2)

emergence of different archaeological theories, including processual archaeology, post-processual archaeology, structural archaeology, and post-structural archaeology, that all tried to explain and interpret material culture and the processes that led to it rather than simply describe its remains (Aldenderfer 1987; Hodder and Orton 1976; Hodder and Hutson 2003; Renfrew and Cooke 1979; Sabloff 2005; Doran and Hodson 1975; Whallon and Brown 1982). Mathematical tools and processes were of course always part of formal analysis in archaeological research, for example, geometrical and arithmetical relations in proportion theory, symmetry analysis, and modular coordination, but it was the so-called "new" mathematics including graph theory, lattice theory, combinatorics, statistical analysis, space syntax analysis, and generative grammars that revolutionized analysis in archaeological research. This paper discusses grammars for the generative specification of artefacts, as the most appropriate in reconstruction. An earlier version was presented in the conference of Computer Applications in Archaeology (Mamoli and Knight 2013).

2 Generative Grammars in the Analysis of Archaeological Artefacts

Grammars have been used in archaeology as a way of classifying and ordering a mass of findings and as a systematic tool of reconstruction that allows possible reconstructions and interpretations (Hodder and Hutson 2003). Archaeological grammars are based on the notion of grammar in language, as described by Lévi-Strauss (1963), where a finite set of rules defines an infinite set of words to produce text and meaning (Clarke and Chapman 1978). This notion is very useful in archaeology, which deals with a massive amount of fragmentary data that once belonged to a "grammar", and is at the core of archaeologists' systematic method of classification and typology. As archaeological research and excavation progresses, and new material evidence come to light, these grammars can be modified to account for the new designs, or the new material can verify hypothesized designs generated by the grammars.

To counterbalance the criticism that formal methods look at form detached from historical, cultural, and social meaning, Clarke has suggested a model of three separate but parallel grammars, a pragmatic grammar to address the relationship of artefacts to the individuals that conceptualize (maker) and perceive (observer) them, a semantic grammar to address the relationship of artefacts to their context, and a syntactic grammar to study the artefacts as geometric entities with a set of attributes. Among these grammar types, syntactic grammars are of particular interest because they address shape and geometric form, the basic fundamental physical variables of material evidence (Clarke and Chapman 1978).

Syntactic grammars are particularly useful in the classification of artefacts and the interpretations of the processes that generate the specific geometry and form of these artefacts because they focus on the geometric properties of artefacts, and they don't require an apriori interpretation of function and meaning to explore systematically diverse ranges of possibilities (Chippindale 1992; Clarke and Chapman 1978; Fekri 1988). Syntactic grammars can be developed with symbols, and therefore are symbolic grammars, or visually with shapes, so they are shape grammars (Gips and Stiny 1973; Stiny 1975). They both use a vocabulary—either of symbols or shapes—and a set of rules according to which the vocabulary is manipulated recursively. The difference is that shape grammars use directly shapes for the computation, as opposed to scripts and codes for the representation of shapes and transformations and operations and therefore are intuitive in the hands (and eyes) of the researcher, archaeologist, and/or architect who is required to reason directly with rules and shapes. In addition, shape grammars work with embedding and as shapes fuse, they provide a mechanism to deal effectively with emergence in spatial design and therefore can become a more effective formal model for visual enquiry.

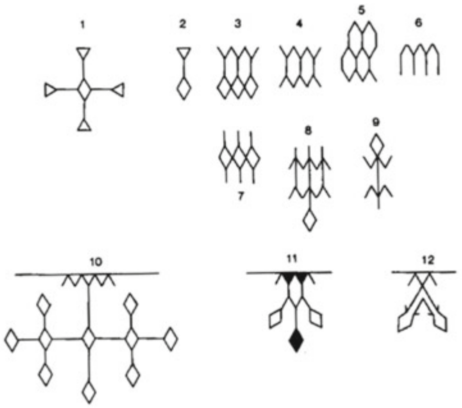
In addition to form, function and materiality can also be embedded in the computation with the use of labels. Labels are markers that are attached to shapes to restrict the application of rules or to add an extra layer of meaning, function, or material to the shapes. Additionally, shape grammars can use parametric shapes, which are defined parametrically according to a set of parameters and a range of valid values to these parameters.

In the analysis of style in architecture and the visual arts, shape grammars can be particularly useful because they provide a code for type and style definition and stylistic change. To develop a grammar for the analysis of style, a set of known designs is analysed, the elements of design or the vocabulary is identified, and the design principles are explicitly encoded in rules.

Symbolic grammars were the earliest examples of applications of grammars in archaeological research (Hodder 1982) and not much later shape grammars were used too (Chippindale 1992; Weissman-Knight 1986).

2.1 A Symbolic Grammar for the Decorative Arts of Nuba Hills in Central Sudan

The earliest example of a symbolic grammar in archaeological discourse is the grammar by Hodder of the decorative arts in the area of the Nuba hills in central Sudan (Hodder 1982). Nuba ornaments appear in the decoration of hut fronts, pots, calabash artefacts, and body painting. The grammar encodes constructively the type definition of Nuba designs by decomposing the underlying motif, the star motif, into a vocabulary of shapes, typically a diamond, a triangle, a chevron, and a line segment, and a set of spatial relationships between these shapes (Fig. 2a). These spatial relations require that the shapes and the relations should be joined at their vertices in oblique angles and not along the sides or the bases of the triangles and the diamonds. The resulting grammar consists of a small set of commands (Fig. 2b) that can compute known and hypothetical designs (Fig. 2c).



a.

Instructions

A. Choose object and part of object (calabash, house, etc.).
B. Choose starting element (word unit).
C. Begin design generation.

Moves	*Possible operations*
1. Move horizontally	At vertices (if possible or oblique angles) CONnect to (R)ight or (L)eft: CONnect after ROTating (in units of 45° or 90°): EXPand size of element: SKIP
2. Move vertically	As 1, moving Up or Down: SKIP
3. Fill	SOLid fill: DESigned fill of the spaces generated in 1 and 2: SKIP
4. Rotate	ROTate whole design so far completed in units or multiples of 90° and redraw: SKIP
5. Expansion, reduction	EXPand or REDuce size of whole design so far completed by 'n' units – a unit being the length of the sides of the A and B elements: SKIP
6. Repeat	REPeat design so far completed *n* times, VERtically or HORizontally, to Right or Left, Up or Down: SKIP
7. Return	RETurn to *n*th step or starting point

b.

Examples

(a) 

Starting point: Choose B_1, ROT(270°), SOL. (Comment: the left triangle is chosen, and obtained by turning B_1 through 270°, and filled.)

1. CON(R) ROT(180°)
2. SKIP
3. SKIP
4. SKIP
5. SKIP
6. REP(1) HOR(R)
7. SKIP

(b) 

Starting point: Choose A, SOL. (Comment: the left lozenge is chosen and filled.)

c.

Fig. 2 The Nuba grammar: **a** schematic representation of Nuba motifs across different media as configurations of the star motif; **b** the rules of the grammar (operations or instructions); **c** two examples of computations (Hodder 1982, Fig. 81 and Table 18)

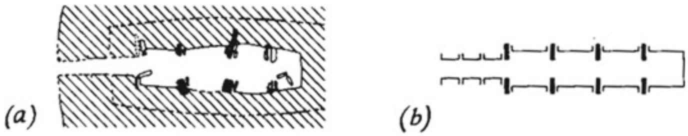


Fig. 3 A megalithic chamber tomb from Calf of Eday Long, Eday, Orkney, Scotland. **a** Plan, **b** regularized plan of the tomb (Chippindale 1992, Fig. 10)

In the Nuba grammar, the rules are used as an experimentation tool, to test the validity of the theory about the underlying structure of the Nuba designs, and the variation that appears among different designs that appear in different media, or among different groups of people. Hodder states that by defining the whole range of possibilities, we define the full richness, complexity, and subtlety of style and design, and we make clear not only the extent of similarities but of differences as well. This contributes to the interpretation of why particular traits are not adopted in local areas. By knowing "the actual among the possible designs", he was able to identify which underlying rules are tribe-specific and which rules are common to all tribes of the Nuba.

2.2 The Orkney Islands Chamber Tombs Shape Grammar

An early example of a shape grammar in archaeological context is the grammar of megalithic tombs in the Orkney Islands developed by Chippindale (1992). The grammar analyses and describes constructively the design and the process of the generation of megalithic, chambered tombs constructed out of flat stone slabs, subdivided into modules by vertical slabs, in a linear arrangement. With six rules that add extra flat stone slabs and vertical slabs and terminate the design, the grammar describes the form of known tombs and the process of their construction (Fig. 3a).

The algorithmic generation of chambered tombs is represented visually with shapes (Fig. 4), but also numerically as a string of numbers that give the sequence of rules applied to compute each derivation. Additionally, the grammar classifies the tombs in schemata according to the different processes followed for their actual construction in a very simple way.

2.3 The Greek Meander Shape Grammar

The shape grammar for the Greek meander motif in Geometric pottery (Knight 1994; Weissman-Knight 1986) is a great example of how the transformation of a shape grammar can produce a related grammar with a related language of designs that can explain stylistic change and transitions from one style to another. As defined

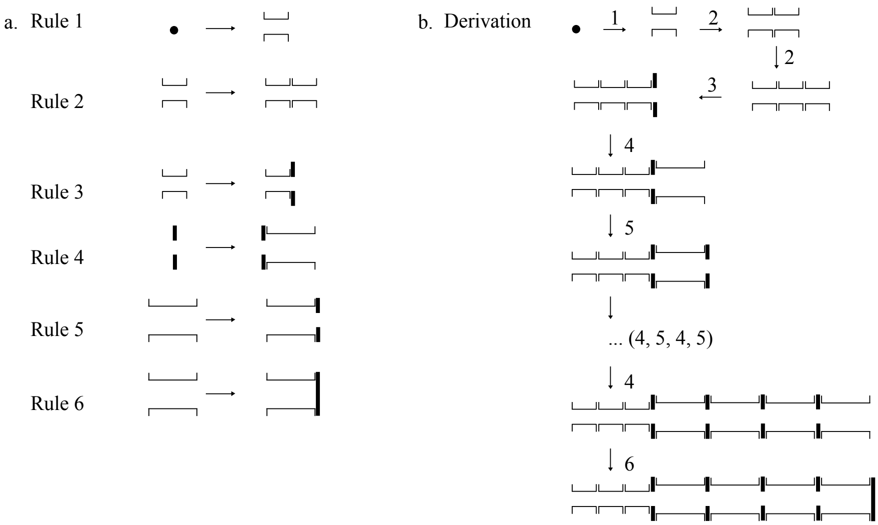


Fig. 4 The shape grammar of the megalithic chamber tombs at the Orkney Islands. **a** Shape rules; **b** derivation of the East chamber at Calf of Eday Long, Orkney, Scotland (author's drawings after Chippindale 1992, Fig. 10c)

by Knight, transformations of a grammar entail three operations, rule deletion, rule addition, and rule change. Rule change happens either with a change in state labels, which affects the sequence the rules are applied, or with a change in spatial labels that affects the way a rule is applied, or by changing the vocabulary of shapes or their spatial relationships. Transformation grammars are valuable in approaching stylistic change between two or more styles; transformations in the grammar explain the transformation of the first style into the second, as well as the transformation of the first style into a variety of other styles, existing, or hypothetical.

The grammar is designed in three stages to capture stylistic change due to temporal and geographic criteria. In the first stage, the grammar consists of the initial shape and two rules (Fig. 6a, b) that generate simple battlement and running meanders of the Early and Middle Geometric pottery (Figs. 5a, c).

In the second stage, the grammar is transformed with rule addition to account for the more complex meanders of Late Geometric pottery (Fig. 6b–d); a stacking rule allows the stacking of meanders, and a rule deletes the overlapping lines to generate double and triple meanders. Additionally, rule change with shifts of 0, ¼, or ½ units account for meanders of different areas (Fig. 6d). In the third stage, the grammar is transformed to account for stylistic change based on the style of different workshops of the same area and same period.

The transformation shape grammar for the meander motif in Greek vase painting is very successful in that it takes a very complex motif, that in Greek art history is usually bypassed as a decorative motif and dissects it into sophisticated stacking

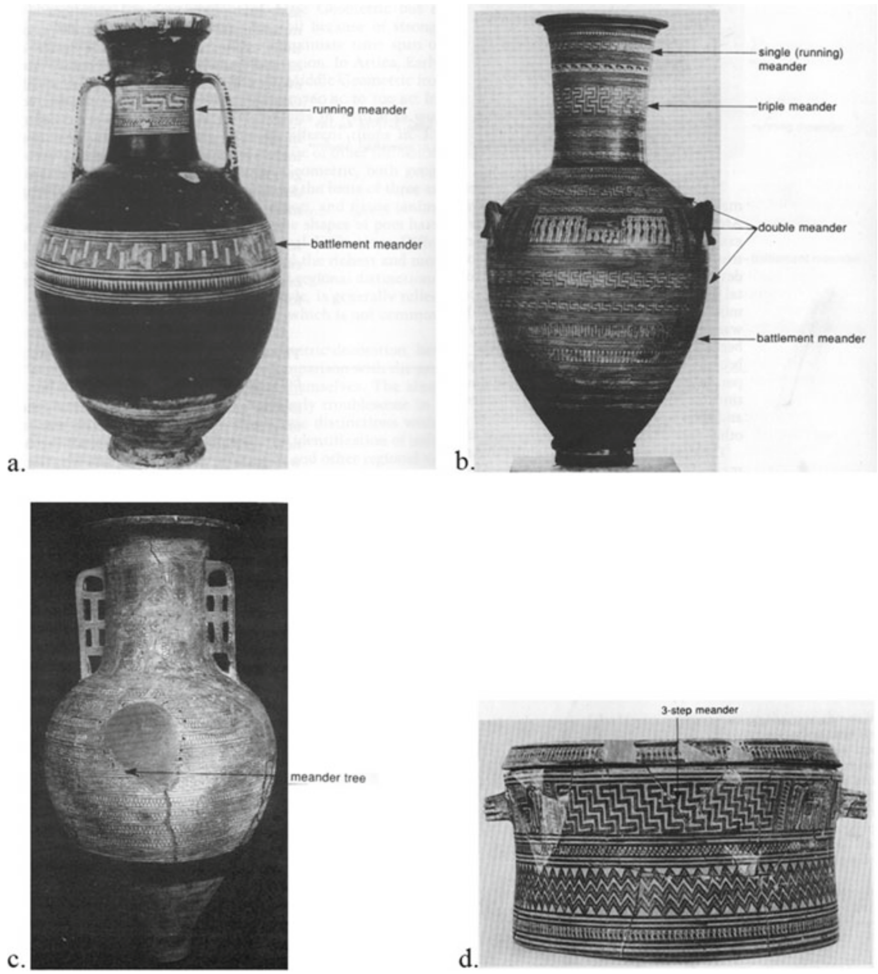


Fig. 5 Geometric pottery. **a** Early Geometric battlement and running meanders, Attica; **b** Late Geometric single, double, and triple running meanders, Attica; **c** Late Geometric meander tree, Rhodes; and **d** Late Geometric step meander, Argolid (Knight 1994, Fig. 6.1 a, c, w, p)

rules that reveal a complex mind in its making. Also, transformation rules reveal and encode the spatial provenance and identity of its makers.

3 Shape Grammars for Building-Type Definition

For the study of a building type, a comparative analysis of existing instances within the same type or class of buildings concludes into the main design features that are

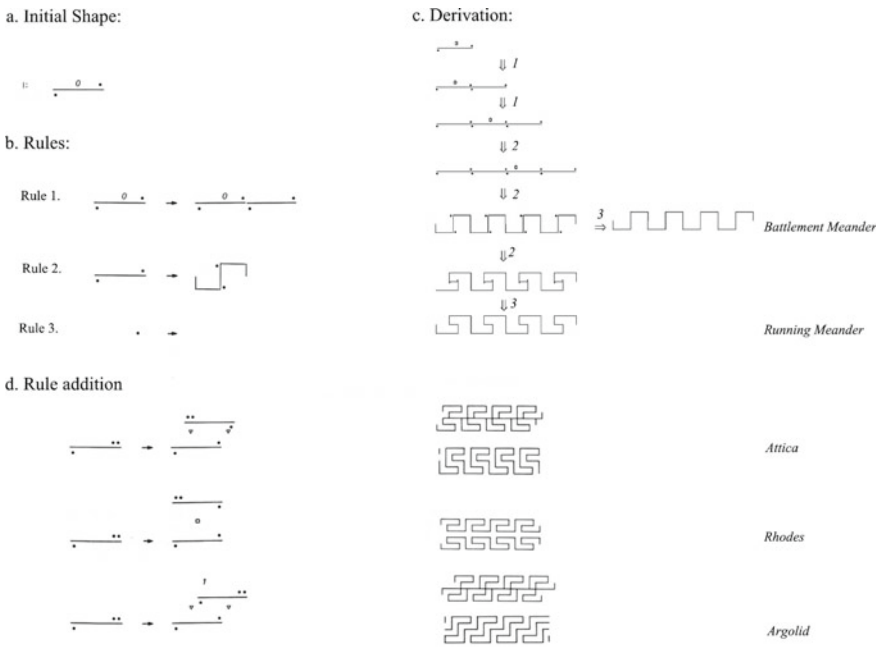


Fig. 6 Transformational shape grammars for the Greek meander. **a** Initial shape; **b** rules that generate the Early Geometric battlement meanders, and the Middle Geometric running meander; **c** derivation of the Early and Middle Geometric meanders; and **d** addition of the stacking rule to generate Late Geometric meanders. Rule change in the stacking rule in terms of ¼, 0, and ½ shifts account for stylistic change between different regions (author's drawings after Knight 1994)

common across the whole corpus. Certain variations can be identified and decoded into schemata that allow us to focus on certain properties. However, depending on the degree of detail we choose to look at, schemata do not account for all variability, as they give the impression that characteristics included in one schema cannot appear in the others, therefore excluding combinations between them. This problem is bypassed with the use of shape grammars. A set of rules is used to decompose its design principles, that can be combined in as many ways as the author defines.

Shape grammars have been used very effectively in a combinatorial way with procedural modelling software in the generation of cityscapes, for example, Pompeiian architecture, the reconstruction of Rome and also Mayan architecture (Müller et al. 2006a, b; Dylla et al. 2008). In these projects, the procedural modelling of the city allows for the quick generation of alternative buildings and cities with components that follow an initial set of rules and therefore are in the same style and type, without evoking an overt sense of repetition. These software applications work primarily in a top-down approach that include the successive subdivisions of surfaces that reach a high degree of detail including sculptural details such as frieze, cornices, and ornamental details like acanthi and rosettes, and they are limited in the sense that they are based on input from the user about the structure of artefacts and their parameters, and cannot account for shapes that emerge in the process.

Here, I present a shape grammar about the building type of the ancient library, which is still not automated and is in plan only. Still, the grammar provides a model for the definition of the building type of the ancient library with mandatory and optional architectural features—some most likely and others less likely to occur—which is an effective and systematic guide in identifying, evaluating, and predicting the architectural form of ancient libraries of diverse scale and monumentality.

3.1 The Library Shape Grammar for Ancient Greek and Roman Libraries

The library grammar (Mamoli 2014, 2015, 2018, 2020) is an extensive parametric shape grammar with archaeological content that tackles a whole building type and tries to define it as a combination of mandatory and optional characteristics that appear with various degrees of probabilities. At the same time, it tries to bridge the gap between state of preservation and reconstruction. The grammar is analogue, and the rules generate reconstruction drawings of ancient libraries. However, the computations are overlaid on the state of preservation drawings and the user of the grammar can try different paths based on the remaining evidence in the archaeological record.

Founded around the Mediterranean and spanning a period of about 400 years, public libraries were important institutions of the Hellenistic and Roman city, but they are not as distinct as other building types, such as the temple. The few examples of libraries—17 known from the ancient literary sources that have been identified with building remains—exhibit a great diversity of scale, urban context, arrangement of spaces, and monumentality. Researchers have tackled the development of the building type of the library across time and space, primarily classifying the libraries in Greek and Roman based on the developments in Roman construction and the multiplication of resources in the Roman imperial period, and also into provincial libraries in the eastern or the western part of the empire by attributing the proportions of the main hall to Greek or Roman traditions. These classifications do not manage to resolve the history of the type of the ancient library; because on one hand, there are too few examples in each category to verify the hypothesis, and on the other hand, there are exceptions in each case that undermine the whole classification. Roman emperors became patrons of art and architecture throughout the empire. Officials in lower ranks aspired to them and imitated them and ideas spread throughout the empire and a closer look at the archaeological record shows that the design trails of Greek architecture, the main hall of the library, and its association to a stoa are preserved in the Roman period and appear along the innovations in the use of Roman concrete that allowed bigger spans and also the embedding of furniture in niches.

Overall, libraries of various degrees of monumentality appeared at equal frequencies throughout the history of the library. At its simplest form, the ancient library consisted of a single room with no architectural features of interior design attached to a stoa. At its most monumental form, the design of an ancient library included a main hall with armaria embedded in niches, the statue of the patron in a focal point—niche or aedicula—and a podium around the room. The main hall was flanked by other auxiliary rooms, and it was preceded by a stoa and in some cases by a peristyle with a courtyard that included exedras and a monumental entry to the complex. These two different versions of libraries as well as all the gradients of monumentality in between acquire an equal representation in the grammar as their design principles are broken down and represented in design rules that a user can compute to generate any of them.

To overcome the conundrum of defining a concrete theory about the building type of the ancient library, the author has developed a parametric shape grammar that summarizes and encodes in design rules the design principles that occur in the 17 cases of known ancient libraries (Mamoli 2014, 2020). The methodology includes three steps: firstly, the bibliographical review of measurements and state of preservation drawings, primarily plans but also sections and elevations where the archaeological remains are at a sufficient height; secondly, the compilation of this evidence in a database that will inform the parameters of the rules of the grammar with a valid range of values; and thirdly, the design of the grammar in two dimensions in plan that consists of 91 design rules, subdivided into 12 stages that roughly correspond to the generation process of a library. The first six generate the layout of the main hall with its interior design, including the podium, the niches, the focal point, the interior colonnade, and the entry. The final six stages generate the general layout of the building or the building complex with the side rooms, the stoas, the exedras, the entrance, and the courtyard if any. The rules include metadata specified by letters of the alphabet that point to each library that shows evidence in the archaeological record for each rule.

The application of the rules generates libraries of diverse sizes and monumentality, known libraries in the corpus as well as hypothetical. The user of the grammar can start with the underlying plan of the state of preservation of a library. The plans have been redrawn to follow the same conventions of representation, to integrate evidence brought to light and documented over time in different drawings, and to be deprived of earlier and later building phases and interventions as well as reconstructions. The user selects and embeds the rules in the building remains, either fully or partially. This process involves a lot of decision-making and speculation as the archaeological fragments can generate different maximal lines, i.e. different lines of maximum length can emerge from connecting shorter lines, with the assumption that the latter refers to fragments of the same structure. For example, two wall fragments might or might not belong to the same room or building, and therefore a variety of maximal lines can be identified. Depending on the degree of preservation, the user can secure, speculate, or eliminate the application of some rules. In cases that the building remains present evidence about an architectural feature, the user can either eliminate or select a rule for application. In cases of rule selection, the user can plug in the actual dimensions of the building components as preserved in the archaeological record in the parameters of the rules and reconstruct the plan of the building. A subcase of secure application of a rule is the partial embedding when there is evidence for the embedding of the rule in the building remains, but only partially with some tweaking. In these cases, an exclamation mark (!) next to the rule denotes that that particular characteristic is case-specific or exceptional. In cases that the building remains show no or little evidence and require speculation, the user can pick the rule that s/he considers more probable based on the other libraries and generates the corresponding building component based on the site specifications and the other parallels. In this case, an asterisk (*) next to the rule denotes that the rule is speculative.

Figure 7 gives the derivations of all known libraries along with the strings of rules that apply to demonstrate how this works. The derivations of the known libraries showcase the diversity of library plans that the grammar can generate: libraries that consist of the main hall and a threshold, libraries that are part of a larger complex, and

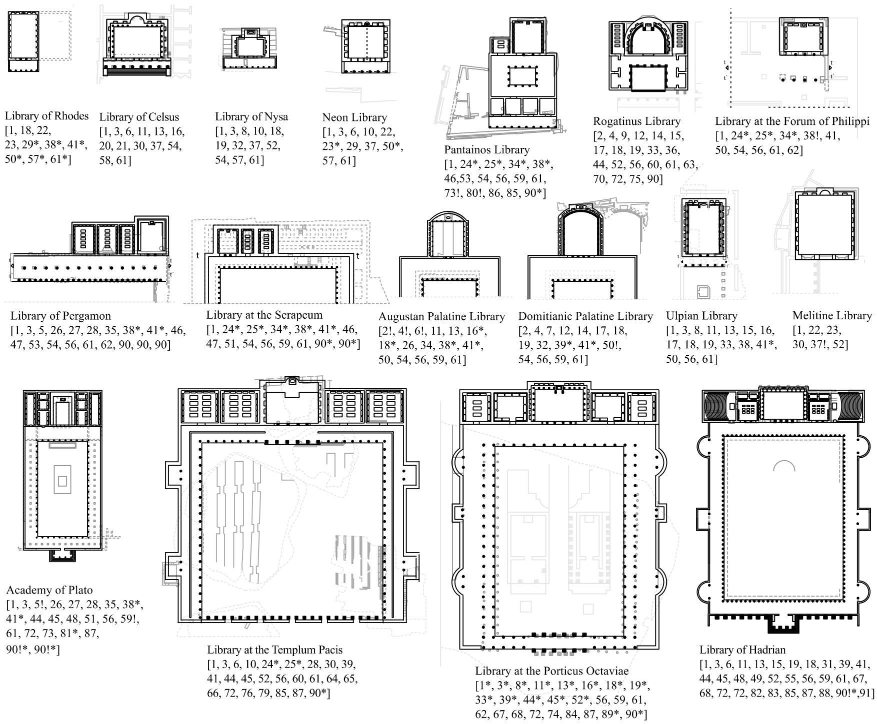


Fig. 7 Computations of all known libraries with the strings of rules apply for their generation. Asterisks (*) denote the speculative application of the rule, and exclamation marks (!) denote a not very precise embedding of the rule in the building remains. The labels tt' indicate that the library is part of a larger complex that the grammar cannot capture. The layouts of libraries showcase the variation in the scale and monumentality of the main hall and the general layout (Mamoli 2020, Fig. 16)

libraries that are a complex themselves. Some of the computations are secure based on extensive building remains, whereas some are speculative due to the lack of evidence (Mamoli 2014, 2015). In cases of well-preserved libraries, scholars securely propose a reconstruction following traditional methods. The grammar is able to re-generate it, which verifies its validity as a tool.

The rules used to generate the 17 known cases, and the frequency with which they apply to generate reconstructions of known libraries are analysed in a frequency analysis graph. The assumption is that rules that occur more often reflect the design principles at the core of the building typology of the ancient library, while rules that occur less frequently reflect stylistic and case-specific characteristics. In this analysis, we exclude rules that apply speculatively and are denoted with an "*". This provides us with quantitative data about the building type of the ancient library, about the mandatory, the most probable and the less probable architectural components in a library, and thus it provides us a probabilistic model for the building-type definition of the library (Mamoli 2018, 2020).

Subsequently, the researcher can keep this data in mind when speculating about the reconstruction of a library, which is partially preserved or not preserved at all. The researcher can choose to apply the rule with the highest probability each time, and thus, the grammar can work as a prediction tool in reconstruction. Figure 8 shows variant hypothetical library main halls by changing the rules of one component each time, the niches, the focal point, the podium, the colonnade, and the entry. They are arranged in rows, one for each variable component, from the more to the less probable according to the frequency analysis of the rules in the derivations of the known libraries.

Ultimately, the grammar can generate hypothetical library plans that can work as a prediction tool and guidance to any excavation by giving hints as to what kinds of evidence one can look for and where. For the unidentified classical buildings, the grammar can work as an evaluative tool—if the rules of the grammar can be embedded in its state of preservation plan, and can generate a possible library plan, then this building is evaluated and interpreted as a possible library. Finally, the formal description of hypothetical libraries can help reconstruct libraries not identified in the record, but known through descriptions in literary sources or dedicatory inscriptions. The grammar can map the parts of the building described in ancient testimonia with corresponding generative rules to provide alternative scenarios of reconstruction and help identify them on site.

4 Conclusions

In the process of reconstruction, archaeologists rely on analogy to parallel artefacts to supply ideas they are missing from a monument under consideration. Shape grammars as a formalism can approach the issue of parallels and precedents in the interpretation and reconstruction process in an explicit and systematic way. The formalism can help archaeologists build up a "thesaurus" of vocabulary and syntactic

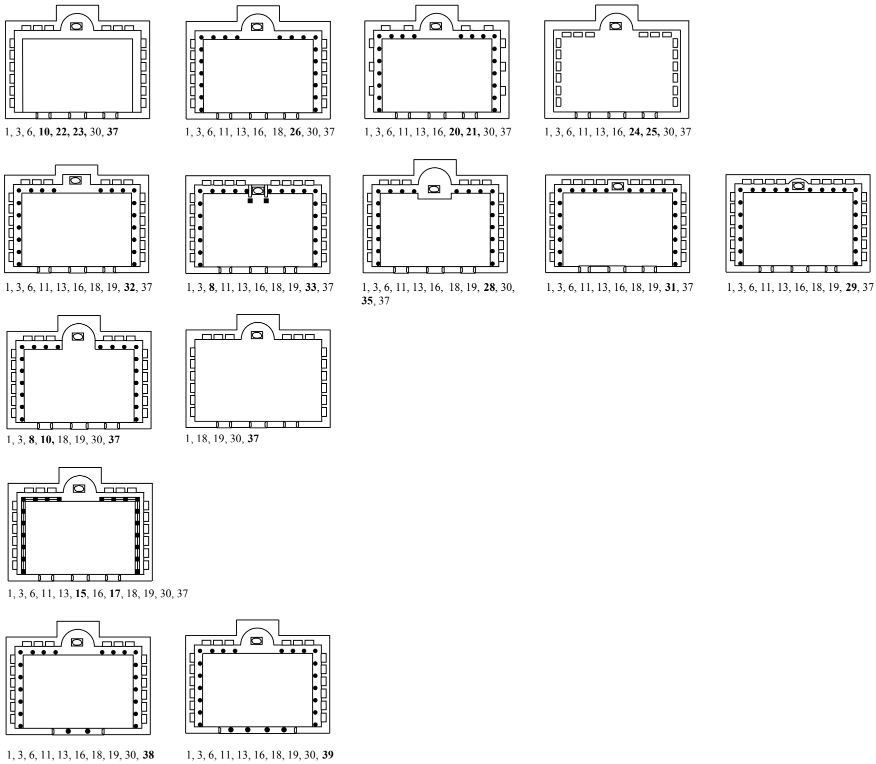


Fig. 8 Hypothetical library main hall plans with the string of rules apply for their generation. Each row illustrates variations of one component (niches, focal point, interior colonnade, steps, main entry), ranked from the most probable to the least probably based on the frequency analysis. The rule responsible for that component is given in bold

relationships, in other words, shapes and spatial relationships, as they are found in the archaeological record. Then, the archaeologist can use them to compute a given problem in variant ways and generate multiple iterations with relatively little effort and time.

Archaeologists do this without computation: they look at spatial relationships in analogous situations and try to apply them to the artefacts under investigation. But the advantage of the formalism is that it includes the vocabulary and spatial relationships systematically. This data can be understood not as a single recipe, but as a recipe with stages, in each of which there are different options. The archaeologist can choose each time a different path in the lattice of options, until s/he achieves an enumeration of different possibilities. The actual and most probable interpretation/reconstruction is certainly among them. Then, it is the responsibility of the informed archaeologist to look at the possibilities, narrow them down based on social and ideological criteria, or criteria based on context and identify the ones that are more probable.

Shape grammars enable the researcher to explicitly articulate statements about the artefact and its underlying configuration in visual rules that are intuitive and descriptive and can be tested against the data. They provide a systematic approach to explore variation and possibilities within a typology or style. This is a powerful tool in the hands of the historian who can identify the actual among the possible. Knowing the range of possibilities, the historian can make a hypothesis and explain and interpret the existence or non-existence of designs among cultures and cultural groups. Their transformations with rule addition, rule deletion, or rule change can produce transformational languages of designs, an excellent guide to explore stylistic change and to enquire of how different styles are related to each other.

Having an enumeration of possible scenarios is of value even in cases where the most probable scenario is obvious. In the event new evidence comes to light, and the probabilities change, the scholar can go back to the possibilities and reevaluate them. Thus, s/he is not entrapped into a possible reconstruction that was suggested by one archaeologist at one given time.

Moreover, embedding design rules into the building remains for the computation of the rules helps to better understand the archaeological record and encourages the archaeologist to look for evidence that might not look important at first sight, or that might be stripped off the building at a later phase. Looking at the design principles that might have generated the building allows the archaeologist to think like a designer, and look not only at the final outcome of the building but also at the design intent of the building that might not be obvious in the way the archaeological record has formed through later interventions or decay.

Also, the visualization of different possibilities has value in reaching a better understanding on the classification of the archaeological record into variable and non-variable remains. Obviously, the variable parts are the ones least preserved that allow room for speculation and the variant application of different rules for their reconstruction.

In conclusion, shape grammar functions as a powerful methodology that helps the archaeologist have consistency in the interpretation process and understand the repercussions of his/her theories in different settings. On one hand, the grammar helps the archaeologist achieve the full enumeration of possibilities according to his/her theories, and on the other hand, it helps the archaeologist understand and visualize his/her theories in different buildings and understand the variation with which the proposed design principles in a style or type appeared.

Regarding the constant check between the existing evidence as documented in the state of preservation drawings and the proposed reconstruction by the application of the rules, recent developments in AI by the Shape Computation Lab at Georgia Tech have proven that vector identification of representations of building remains can be automatically identified and reconstructed (Economou et al. 2019), or at least that the computer can identify building remains and propose which ones could co-belong to the same wall or artefact and fill in the gaps by applying the appropriate rule automatically.

The specific challenge that rises here is the deployment of the formalism in the reconstruction of the initial state of an artefact that is given only fragmentarily. Here, the original data that need to be deconstructed in elements and spatial relations survives only partially and the rules have to be of another sort. There are many ways to complete a picture, and the various ways that architects or archaeologists complete the picture of a representation of a building in some initial state can differ dramatically and can produce different results. It is clear as well that the ways a building or an artefact can be completed based on some evidence can differ dramatically with respect to the kind of building or artefact it is, and our prior engagement and understanding of it.

Finally, the grammar as a computational tool made by the archaeologist incorporates the subjectivity of the archaeologist in the acceptance or rejection of current theories and the way s/he describes, interprets, and evaluates the archaeological record. The data exist, and it is up to the researcher to deconstruct these data in specific elements and relations, and design rules and apply them to produce the original data plus additional descriptions that can substantially enhance and/or challenge the original interpretation of the work. In the end, the grammar encodes in visual rules the design principles that the archaeologist has already defined, accepted, or rejected and thus cannot be considered as an objective methodology, stripped of any layer of interpretation. However, it is a systematic way of visualizing and approaching variation that can help the archaeologist understand the full spectrum of possibilities within his/her own theory. As new evidence comes to light, or as the interpretation model changes, the grammar can be modified to reflect the changes and thus can be used again for the investigation of the new ideas.

References

Aldenderfer, M. S. (1987). *Quantitative research in archaeology: Progress and prospects.* Newbury Park, CA: Sage.

Chippindale, C. (1992). Grammars of archaeological design. A generative and geometrical approach to the form of artifacts. In C. Peebles & J-C. Gardin (Eds.), *Representations in archaeology* (pp. 251–276). Bloomington: Indiana University Press.

Clarke, D. L., & Chapman, R. (1978). *Analytical archaeology* (2d ed.). New York: Columbia University Press.

Doran, J., & Hodson, F. R. (1975). *Mathematics and computers in archaeology.* Cambridge, MA: Harvard University Press.

Dylla, K., Frischer, B., Müller, P., Ulmer, A. & Haegler, S. (2008). Rome Reborn 2.0: A case study of virtual city reconstruction using procedural modeling techniques. *Computer Graphics World, 16*(6), 62–66.

Economou, A., Ligler, H., Park, J., & Hong, T.C. (2019). Shape Machine: A primer for visual computation. In *Cultural DNA Workshop 2019: Computational Studies on the Cultural Variation and Heredity* (pp. 79–102), KAIST, Daejeon, South Korea, Korea, June 2019, Advanced Institute of Science and Technology.

Fekri, A. H. (1988). Prolegomena to a grammatical theory of lithic artifacts. *World Archaeology. New Directions in Palaeolithic Archaeology, 19*(3), 281–296.

Gips, J., & Stiny, G. (1973). *Aesthetics systems, Pb.* Stanford: Stanford University Computer Science Department.

Hodder, I. (1982). *Symbols in Action. Ethnoarchaeological studies of material culture.* Cambridge; New York, NY: Cambridge University Press.
Hodder, I., & Hutson, S. (2003). *Reading the past: Current approaches to interpretation in archaeology* (3rd edn.). Cambridge England; New York: Cambridge University Press.
Hodder, I., & Orton, C. (1976). *Spatial analysis in archaeology, New studies in archaeology.* Cambridge; New York: Cambridge University Press.
Knight, T. W. (1994). *Transformations in design: A formal approach to stylistic change and innovation in the visual arts.* Cambridge; New York: Cambridge University Press.
Lévi-Strauss, C. (1963). *Structural anthropology* (2 vols.). New York: Basic Books.
Mamoli, M. (2014). *Towards a theory of reconstructing ancient libraries.* Ph.D. Dissertation, Georgia Institute of Technology, USA.
Mamoli, M. (2015). Library grammar: A shape grammar for the reconstruction of fragmentary ancient Greek and Roman libraries. In B. Martens, et al. (Eds.), *ECAADe 2015 Real Time—Extending the Reach of Computation. Proceedings of the 33rd International Conference on Education and Research in Computer Aided Architectural Design in Europe* (Vol. 1, pp. 463–470) (Vienna).
Mamoli, M. (2018). Shape grammars as a probabilistic model for building type definition and computation of possible instances: The case study of ancient Greek and Roman libraries. In *Design Computing and Cognition '18*, *Proceedings of the 8th International Conference Design Computing and Cognition, Politecnico di Milano* (pp. 499–518), Lecco, Italy, 2–4 July 2018, Springer.
Mamoli, M. (2020). A shape grammar for the building-type definition of ancient Greek and Roman libraries. *AIEDAM, 34*(2) (Accepted 1/7/2020).
Mamoli, M., & Knight, T. (2013). Reconstructing fragments: Shape grammars and archaeological research. In G. Earl, et al. (eds.), *Archaeology in the Digital Era. e-Papers from the 40th Conference on Computer Applications and Quantitative Methods in Archaeology* (pp. 888–896): Southampton, 26–30 March 2012. CAA 2012 (Amsterdam).
Müller, P., Vereenooghe, T., Wonka, P., Paap, I., & Van Gool, L. (2006). Procedural 3D reconstruction of Puuc buildings in Xkipché. In *Eurographics Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST).*
Müller, P., Wonka, P., Haegler, S, Ulmer, A., Van Gool, L. (2006). Procedural modeling of buildings. In *ACM SIGGRAPH 2006/ACM Transactions on Graphics (TOG).* ACM Press.
Renfrew, C., & Cooke, K. L. (1979). *Transformations: Mathematical approaches to culture change.* New York: Academic Press.
Sabloff, J. (2005). Processual archaeology. In C. Renfrew & P. G. Bahn (Eds.), *Archaeology: the key concepts* (pp. 212–219). London; New York: Routledge.
Stiny, G. (1975). *Pictorial and formal aspects of shape and shape grammars and aesthetic systems.* CA: Los Angeles.
Weissman-Knight, T. (1986). Transformations of the meander motif on Greek geometric pottery. *Design Computing, 1*(1), 29–67.
Wescoat, B. (2019). Architectural documentation and visual evocation: Choices, iterations, and virtual representation in the sanctuary of the great gods on Samothrake. In Sapirstein & Scahill (Eds.), *New directions and paradigms for the study of Greek architecture: Interdisciplinary dialogues in the field* (pp. 305–321). Leiden: Brill.
Whallon, R., & Brown, J. A. (1982). *Essays on archaeological typology, Kampsville seminars in archeology.* Evanston, IL: Center for American Archeology Press.

Alternative Design Strategies for Building Façade Screens

Jin-Ho Park

1 Introduction

The geometric patterns of the symmetry groups appear in a variety of traditional window screens and building façades regardless of region and culture. At a close look, simple geometric motifs are recursively copied to generate patterns over the part or the entire building façade. Occasionally, the final design appears to be complicated, visually dominating the façade of the building. The regular repetition of basic motifs relies on the symmetric principles, regulating its overall composition.

The design of building façade is significant because its influence on people is evident in cityscapes. If the building façade is intriguing, then it affects public sensibility. Dynamic building façades may reduce dull feelings (Ellard 2015). Designing building screens or shading devices added to the façade is a crucial element to produce evident effects on indoor thermal comfort, lighting conditions, ventilation, and energy use. Different screen systems, which particularly play a significant role in a desert region due to the extremely harsh weather, have been used in different regions.

Mashrabiya is a crucial building screen that best represents aesthetic and functional purposes in the region. It is a unique and traditional sun-shading device in the Middle Eastern areas and has been used throughout the middle ages until the mid-twentieth century. Mashrabiya is used as a response to the extreme weather of the region, which is hot and dry with intense sun and low humidity. The weather conditions require special solutions to adopt local architecture. Consequently, some passive architectural techniques mitigate these conditions.

The notion of Mashrabiya has recently reemerged with the advancement of technology and subsequently revived in many contemporary high-rise buildings.

J.-H. Park (✉)
Department of Architecture, Inha University, 100 Inha-ro, Michuhol-gu, Incheon 22212, Republic of Korea
e-mail: jinhopark@inha.ac.kr

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_8

Demanding ever-increasing needs for energy efficiency and ecological buildings, the potential application of Mashrabiya has continued to grow in contemporary building practice. A variety of Mashrabiya-inspired screens may be further explored to better control microclimates of buildings.

This paper first introduces the notion of Mashrabiya with regard to cultural, functional, and aesthetic issues. Then, its contemporary applications in architecture are exemplified. Owing to the deeply embedded forms and patterns of Mashrabiya in the symmetry principles, particularly the wallpaper group, the principle is briefly reviewed. Finally, the principle is further employed to test new design possibilities for building screens.

2 Mashrabiya: A Unique Device for Building Screens

Mashrabiya was used throughout the middle ages until the mid-twentieth century. Its industry grew and thrived during the Mamluk era and then extensively spread out in different Arab regions in the Ottoman era (Maspero 1974). Mashrabiya has different styles, each reflecting the culture and identity of the region. As a form of window, Mashrabiya covers openings in buildings with decorative patterns. The window typically comprises a projecting oriel window made of wooden latticework with geometric patterns. It is protruded from the outer wall mainly on the second or high level of a building.

Different types of Mashrabiya are mostly found in urban populated areas across the Middle Eastern and North African countries. The differences are mostly a result of the different components, materials, and crafts as well as the cultural and social values of each country. For example, some Mashrabiya in Yemen are made of stone, whereas most Mashrabiya are made of the lattice-like assemblage of small wood pieces. The Mashrabiya is either flushed with the wall surface or mostly jutted over the street and can be made as an extension of a room. The area is filled with cushions where people can sit comfortably to enjoy the view over the street while facing and feeling the wind breeze directly (Abdelkader and Park 2018) (Fig. 1).

Mashrabiya has an environmental, social, and aesthetic function. First, it works as an environmental mediator to break the direct sunlight, adjusts the temperature, control the humidity, and provides natural ventilation. Mashrabiya also prevents unwanted direct solar gain to the building in summer and allows solar gain to keep the building warm in winter. Accordingly, different sizes of openings and the porosity of the lattice design influence internal airflow and light filtering.

The other key feature of Mashrabiya is a social function that addresses the needs of the Middle East people and the main teachings of the Islamic religion. Mashrabiya provides privacy and view. In particular, old towns were planned with exceptionally narrow alleys because most people had to walk around at that time. Buildings line up along the alleys and face each other, making privacy a main concern for the people living in these buildings. Moreover, the buildings are extremely close to each other;

Fig. 1 Interior image of a traditional Mashrabiya (Image from https://www.autodesk.com/redshift/al-bahr-towers/)

therefore, when windows are opened, one can see all the movements of opposite neighbors.

Mashrabiya is a perfect solution to solve these issues. On the one hand, it allows for occupants to overlook the streetscape and the outer activities without disturbance from the outside. Mashrabiya also secures the view of the outsiders into the house. On the other hand, wooden patterns make it difficult for passersby to observe who is behind the Mashrabiya, allowing the occupants to see outside. It also permits women to overlook the inner courtyard in the upper floors and keep an eye on the activities without being observed by visitors hosted by men in the same house.

More importantly, Mashrabiya has an unusual effect on the aesthetics of the house façade with its unique geometrical patterns, which are deeply rooted in the Middle Eastern culture. Through its patterns, local identity is added to each country due to the varying wooden lattice designs from region to region (Sidawi 2012). Owing to cultural and religious reasons, Islamic patterns utilize geometric or floral forms. Given that human beings usually interpret the words "work of The Creator," its representation is forbidden because anything that exists will fall under the category of imitating God (Ekhtiar and Moore 2012).

The geometric patterns of Mashrabiya appear to have infinite ways. The pattern expands neither with a start nor with an end. Moreover, the pattern relies on two-dimensional plane symmetry group, which is called the wallpaper group. The patterns are also shaped with combinations of repeated geometric motifs to form intricate patterns.

3 Contemporary Mashrabiya

The cultural assets provide pedagogical guidance for the lessons learned and the finest practice to guide future work. Mashrabiya still has a lasting influence throughout the Islamic architecture and has been extensively revived in the practice of contemporary architecture. Although numerous different examples are ubiquitous, two contemporary building screens inspired by traditional Mashrabiya are presented. These designs share the similar aesthetic and functional principles (Abdelkader and Park 2018).

Doha tower is the first example of the wallpaper-patterned structure. The design of this tower is unique, showing a kind of Mashrabiya integrated as its outer skin. Architect Jean Nouvel integrated the notion of Mashrabiya into the screen system in the building with unique geometric patterns. The system is neither an operable nor an interactive screen that reacts according to the changing climate condition. It is a fixed four-layered aluminum screen that simply covers the entire façade acting as a brise-soleil. The curtain wall is placed behind the shading screen. A gap, which exists between the curtain wall and the screens, is accessible for maintenance walkways.

Considering that Doha city is known for its hot weather and high sun glare, the screen functions as protection from heat and direct sunlight and provides interior shading. Geometrical patterns that vary in scale clad the window wall entirely. According to solar directions, the densities of the motif layers are designed differently on the window wall. The opacity is also significantly high on the sides facing the direct sun. The device allows transmission of natural light in a fairly scattered manner.

The screen appears to be extremely complicated when observed from a distance (Fig. 2). However, an in-depth examination is considered to be one of the most simple processes in forming the final screen design. Individual analysis of layers shows that

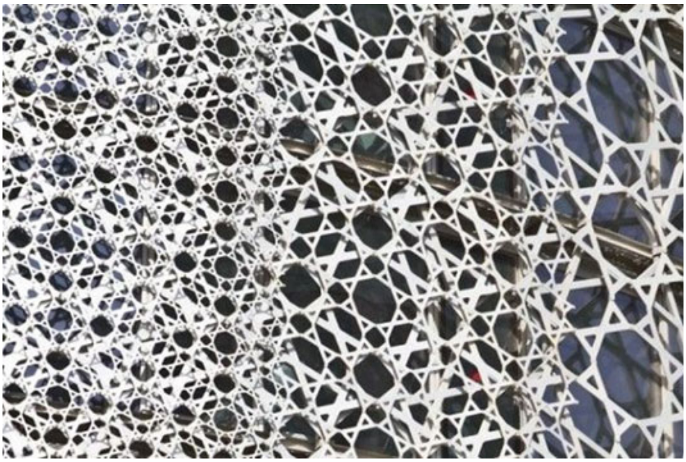


Fig. 2 Photograph of the Doha tower screen (Image from https://thetowerinfo.com/buildings-list/doha-tower/)

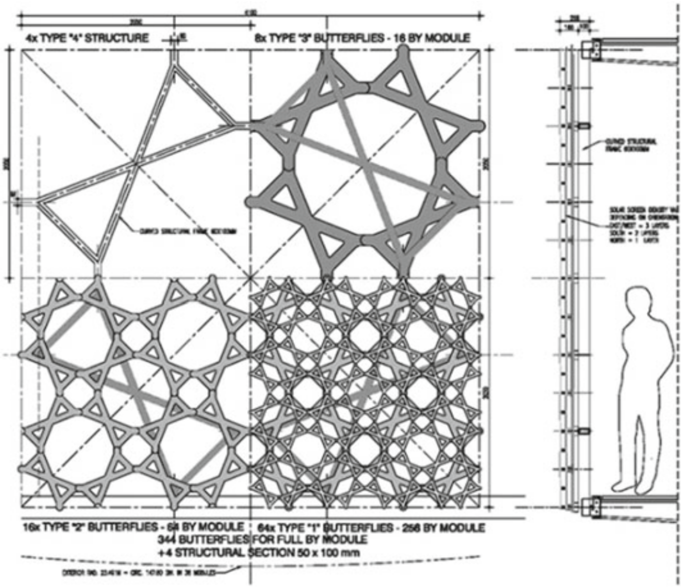


Fig. 3 Sun screen of Doha Tower (https://archnet.org/sites/15150/media_contents/113249)

four different layers of the same design but different scaled patterns are overlapped together.

A unit motif of the pattern appears to be a triangular shape (Fig. 3). The basic pattern shows a rotated, reflected, and translated motif, which forms a symmetry group of p3m1. The same pattern is then reduced consecutively and overlapped in two, three, or four layers. Each design is chosen as a double-layer screen in response to the orientation of the sun and direct light intensity throughout the year. Each aluminum layer is tied up together with thousands of rivets, bolts, and spacers that hold the screen together.

The second example is the Al Bahar Twin Tower designed by Aedas architects. The building is wrapped with an adaptive sun shading screen. The screen is a dynamic and sun-responsive shading screen set approximately two meters apart from the typical curtain wall. Walkways exist in between the screen and the curtain wall and are accessible for maintenance.

The intelligent skin works as an envelope that wraps three-quarters of the building's façade facing the south, east, and west directions. The screen comprises a series of semi-transparent PTFE panels to facilitate subtle diffusion of the transmitting light. These panels are folded and unfolded in response to the orientation of the sun and intensity of direct light, covering part or the entire façade to block the majority of the harsh desert sunlight. Even when fully closed, a certain amount of light can penetrate the interior rooms. The entire screen is programmed to react to the sunlight according to the movement of the sun throughout the day (Cruz 2016). The screen closes to cover the window when excessive heat and sunlight are received.

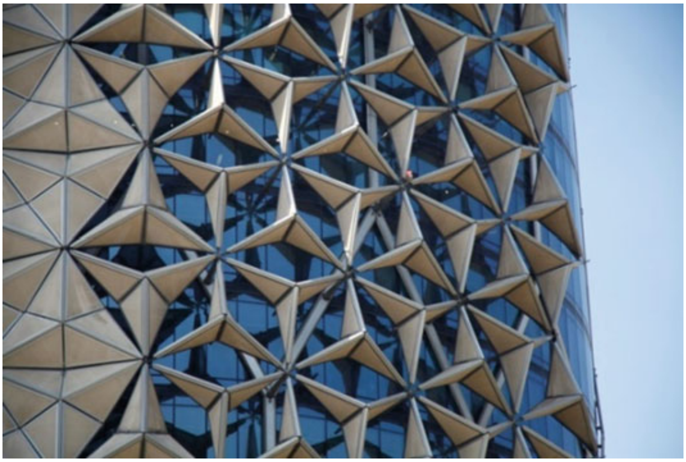


Fig. 4 Responsive screens of the Al Bahar Twin Tower

Similarly, the windows are opened to receive natural light when it is excessively cold or dark. Each solar panel forms a triangle unit motif, which is embodied in a hexagonal responsive and operable unit. The basic pattern of this panel forms a symmetry group of p3m1 (Fig. 4).

Mashrabiya has been reinterpreted in contemporary architecture in diverse ways with considerable popularity. Instead of the traditional wooden lattice, Mashrabiya uses advanced materials in different scaled projects. Nevertheless, the pattern strongly relies on the symmetry principle. Diverse porosities by adopting irregular spaces and small and large openings ensure views from inside to outside while retaining the required privacy and also mediate the direct sunlight. Instead of partially covering the walls in traditional houses, Mashrabiya now acts as a building screen that covers the entire building.

4 Wallpaper Group of Symmetry

The geometric patterns of the Mashrabiya are created by repeating geometric motifs according to the symmetry of the wallpaper groups. The formal definition of the wallpaper groups of symmetry is briefly reviewed to understand the systematic structure and order of the patterns and generate diverse repeated designs that are applicable for building screens.

Two types of plane group symmetry are available in two dimensions: the finite and the infinite groups. The former is called the point group, where isometric transformations occur in a fixed point or line. In the latter, isometric transformations occur under unidirectional translation, namely the frieze group, and two directional translations called as the wallpaper group (Armstrong 1988; Park 2000, 2018).

Herein, only the wallpaper group is considered. Taking a motif as a basic unit, it is rotated and reflected and recursively translated horizontally and vertically. A wallpaper pattern is created by repeatedly deploying a basic motif in a plane. A total of 17 distinct plane symmetry groups are then classified (March and Steadman 1971). The classification is determined by four isometric motions of the plane, such as translations, reflections, glide reflections, and rotations. Isometry is meant to be the process of moving geometric object from one place to another without changing its size or shape. The motion is called an isometry when a geometric figure is transformed to preserve distances between any two points.

Translation refers to moving objects in a fixed distance without changing these objects. Rotation turns an object around a fixed point by half-turns, 120°, 90°, and 60° turns. A reflection is a transformation that flips an object along an axis. The axes can be horizontal, vertical, or at some angle. A glide reflection is a mirror reflection across an axis and a translation along the axis (Martin 1982).

Wallpaper groups are denoted with notation derived from crystallographic notation. Each notation of the wallpaper group comprises four symbols. The groups start with the letter either p or c, which means primitive or centered cell, respectively. A total of 15 cases out of 17 distinct plane symmetry groups are described with respect to primitive cells. By contrast, only two 2 out of 17 distinct plane symmetry groups are described in relation to centered cells.

The next letter "n" represents the type of rotation: 1, 2, 3, 4, or 6. Herein, 1 includes no rotational symmetry. The next letter m, g, or 1 represents mirror, glide reflection, or none, respectively. An "m" at the place of third in the group name indicates a reflection symmetry perpendicular to the x-axis. Meanwhile, a "g" indicates a glide reflection symmetry perpendicular to the x-axis. Finally, a "1" indicates no line of either type. Therefore, a wallpaper group will have rotational symmetries, which include one of 180°, 120°, 90°, or 60°. A wallpaper group can also have reflection and glide reflection symmetries. For example, the group name P31m contains rotations of order 3, that is, a 120° turn. A total of three sets of parallel reflection axes are found in the three sides of an equilateral triangle and are inclined at 60° to one another.

The wallpaper patterns are defined by repeatedly translated copies of basic lattices. Five different lattice units, where each has a different geometric shape, such as square, hexagonal, rectangular, rhombic, and parallelogram, are available. These units are employed to generate and identify a symmetry type of other patterns. Table 1 summarizes that 17 plane symmetry groups can be tabulated by means of their notation and isometries.

5 Design Alternatives for Façade Screens

The screen designs can be diversified with respect to the symmetry of the wallpaper group. Ways of applying the principle, such as strategically deforming the motifs or overlaying different symmetries in different layers, were methodically expanded to potentially inspire better screen ideas.

Table 1 Seventeen wallpaper groups with their distinctive isometries

Wallpaper groups	Lattice unit	Rotation (Angle)	Reflection	Glide reflection	Note
p1	Parallelogram	x	x	x	
p2	Parallelogram	0 (180°)	x	x	
pm	Rectangular/square	x	o	x	
pg	rectangular/square	x	x	o	
cm	Rhombic/square	x	o	o	Glide reflection by centered cell
pmm	Rectangular/square	0 (180°)	o	x	
pmg	Rectangular/square	0 (180°)	o	o	
pgg	Rectangular/square	0 (180°)	x	o	
cmm	Rhombic/square	0 (180°)	o	o	Glide reflection by centered cell
p4	Square	0 (90°)	x	x	
p4m	Square	0 (90°)	o	o	
p4g	Square	0 (90°)	o	o	
p3	Hexagonal	0 (120°)	x	x	
p3m1	Hexagonal	0 (120°)	o	o	
p31m	Hexagonal	0 (120°)	o	o	
p6	Hexagonal	0 (60°)	x	x	
p6m	Hexagonal	0 (60°)	o	o	

Consider an extremely simple condition of a building façade facing southbound. The building has double-skin windows, where the window screen is separated approximately a few centimeters from the glass. A glass window of a building façade, where the arrangement of the motif will become a façade screen as a double skin, is assumed to be covered. The design attempt includes the addition of new window screens to allow and filter the natural light through different window screen designs. The designs will be limited to be as simple as possible to exclude mechanically operational device in this experiment.

Suppose a black and white geometric motif is chosen. The motif is the most basic and smallest unit of a pattern. The chosen motif is repeatedly arranged as a pattern according to the symmetry of the wallpaper groups. Infinite ways in which a motif might be chosen and arranged, transformed, and overlaid for a screen structure are available (Fig. 5).

First, the symmetry of the wallpaper groups can be strictly applied to generate distinct window screens, wherein each screen corresponds to a specific plane symmetry group. After arranging each motif according to the symmetry principles, architects can then manipulate and transform the motifs in a principled way according

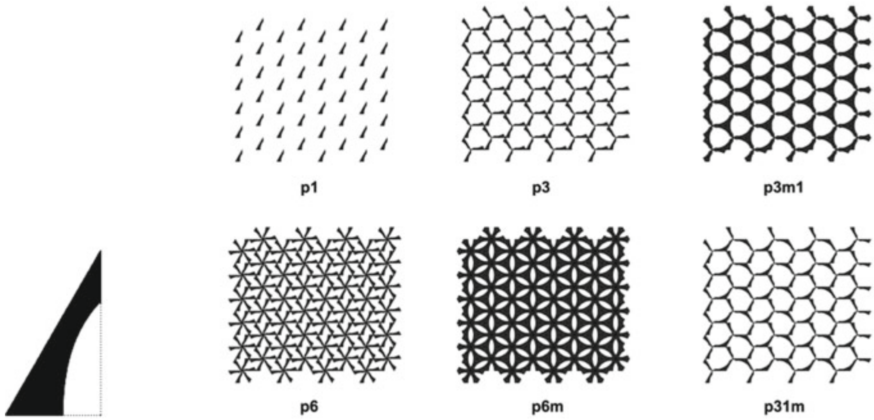


Fig. 5 Left. A motif to be arranged; Right. Some pattern examples using hexagonal lattice

to the functional and aesthetic needs of a building. Finally, different ways of overlapping and binding the symmetries of the wallpaper groups in a single screen design can be employed.

Some screens clearly exhibit which symmetry principle is applied in a pattern, whereas others are visually compelling but less connected to their flat representations of symmetry. The final patterns may possibly be ambiguous compared with those where the symmetry principles are strictly used. Instead of arbitrarily designing the screen, part of the joy of designing window screens in such a principled way is balancing deliberate structure and artistic appeal.

5.1 Alternative 1

A total of 17 distinct symmetries of the wallpaper group in the plane are simply employed to create different patterns for a building screen. Repeated motifs are engraved to cover the entire façade. The motif can also be filled with or outlined with different scales, such as that of Jean Louvel's Doha Tower project. Given that the final screen design will significantly influence the external appearance of buildings and the comfortable living environment of their occupants, selecting an appropriate motif after thoroughly testing various designs may be crucial. Architects can then choose any design most appropriate to their functional and aesthetic needs (Fig. 6).

Fig. 6 Some examples of screen configurations simply using the symmetry of the wallpaper group (Constructed by Seoung Beom Park)

5.2 Alternative 2

The spatial deformation of the motifs is integrated into a planar façade. Different depths or thickness of embossed motifs may be arrayed according to the symmetry of the wallpaper groups. The motifs may also be bent, twisted, or deformed with certain rules.

Some functional and aesthetic issues may be considered in determining the designs of the motifs. These issues will affect the porosity of the screen, which will eventually influence the solar gain, glare, and even the airflow. Such approaches would also develop various spatial effects of an interior atmosphere; wherein natural light can be inserted into the room differently during the daytime. Another effect is the silhouettes from indoor light at night time.

When the motifs are deformed, the screen will provide the viewer with a complex structure of visual experiences in different perspectives. Although the façade is viewed from the front, the screen looks similar to that of alternative 1. However, by looking at the screen from different angles, each design may appear as dynamic spatial images with the deformed motifs (Fig. 7).

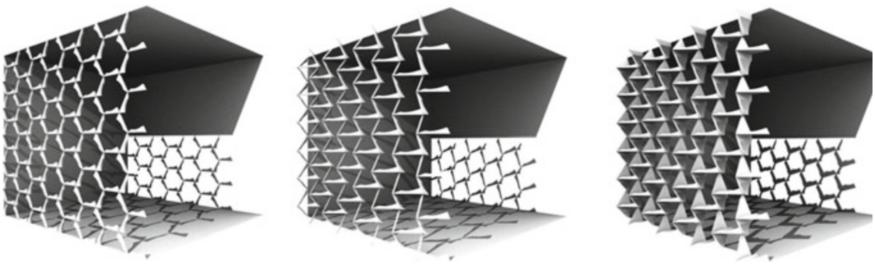


Fig. 7 Some examples of the deformed screens using a hexagonal lattice

5.3 Alternative 3

In this section, experimenting with ways of overlapping and offsetting various layers of the screen is attempted. Although some symmetries of the wallpaper group can accommodate straightforward patterns, in which a simple motif is threaded according to the symmetry principles, others can be overlapped to generate a rather ambiguous or complex pattern of the window screen (Fig. 8).

When a simple motif is directly arranged to form a screen, the final patterns may possibly appear to be homogeneous to the observer. In another way, mapping different types of the symmetries of the wallpaper groups onto each other or hierarchically overlaying them will create remarkably complicated geometric patterns for the screen. Herein, a few layers of patterns are lined up according to its symmetries. By contrast, the random overlay is also possible, but rigorous ways of overlapping symmetries of the wallpaper groups for a façade screen are limited.

Overlapped screens can be detached or attached to influence ways of daylighting. The offset can be determined by testing the airflow or lighting in the building. Multiple overlaid screen changes the way natural light passes through the screen, thereby also changing the amount of solar gain within the building.

With a close look, the intricacy of the overlapped layering becomes complicated, providing the façade screen manifold layers of textural outlooks. The constructed façade will serve as multi-layered filtration membranes acting as an effective daylighting and shading device to shield harsh or direct sunlight (Fig. 9).

This approach can enumerate infinite numbers of interesting outputs for the patterns. Taking the motif outlined similar to that of Jean Louvel's Doha Tower project as an example to increase the porosity of the screen. Each layer of the screen will be designed in terms of different symmetries of a particular wallpaper group and then stacked in a hierarchical way while offsetting in some distances. Although a

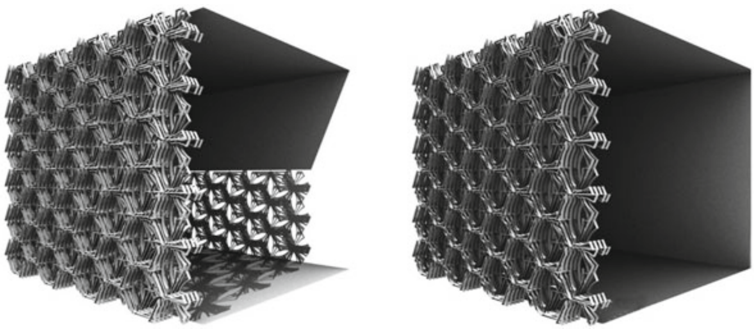


Fig. 8 Some window screens during the winter (left) and the summer (right) where the symmetries of wallpaper group, including p6m, p31m, p3m1, p6, p3, and p1, are overlapped with an outlined motif. From the exterior, the building screen shows subtle undulation caused by overlapping a series of symmetry layers

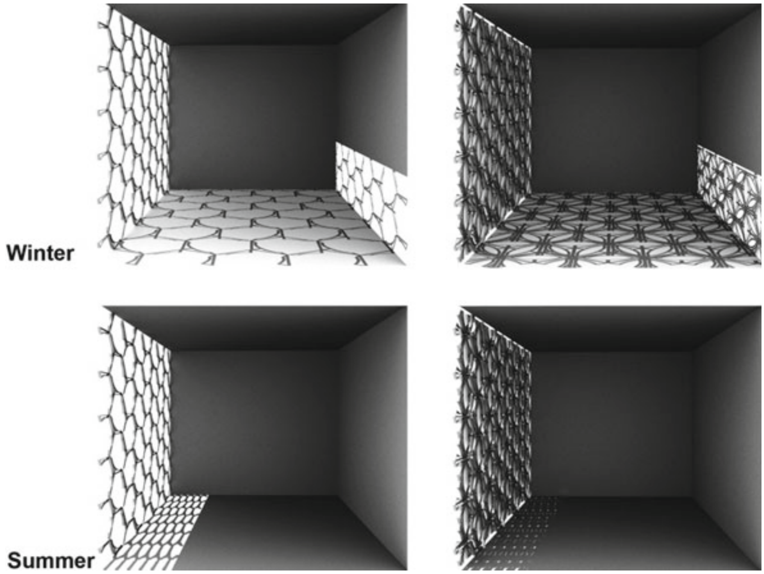


Fig. 9 Interior views of a room during the winter and the summer when a single layer is used (left) and two layers are overlapped (right)

design showing a step-by-step procedure may not be the best illustration, this design is developed to clearly display the final look.

6 Summary

A motivating design model is suggested for the development of repeated and regulated building screens. Herein, the symmetry of the wallpaper groups served as a key principle for generating a variety of screen patterns. The principle can be used to understand repeated patterns in the plane and more importantly, to generate new designs.

Historic Mashrabiya was briefly reviewed with regard to its social, cultural, and sustainable functions. Although its functions have been slightly modified over time, its traditional geometric patterns remained as one of the main characteristics that provide the unique identity of the region. Mashrabiya has been revived in contemporary architecture and further utilized as an integral part of practices for the development of sustainable architecture.

The repeated regularity of the geometric patterns of Mashrabiya relies on the symmetry of the wallpaper groups. Therefore, the symmetry of the wallpaper groups was briefly introduced and then employed to test synthetic window screens. The wallpaper groups may serve as a tool for unlimited creativity for the screen designs in the hands of the architects, making it possible to design a variety of façade structures.

In addition, ways to deform each motif or overlap different patterns of screens to generate different depths of the screens were tested. A simple design clearly exhibits the underlying wallpaper symmetries, whereas others are visually intriguing. Thus, the method has enormous potential for generating a diverse range of different designs and applying into sustainable screen designs of architecture.

References

Abdelkader, R., & Park, J.-H. (2018). Sustainable building facades: modern usages of the traditional Mashirabiya. *Open House International, 43*(2), 69–76.

Armstrong, M. A. (1988). *Groups and symmetry*. New York: Springer-Verlag.

Cruz, P. (2016). *Structures, and architecture: Beyond their limits*. Boca Raton, FL: CRC Press.

Ekhtiar, M., & Moore, C. (Eds.). (2012). *Art of the Islamic world*. New York: The Metropolitan Museum of Art.

Ellard, C. (2015). *Places of the heart: The psychogeography of everyday life*. New York: Bellevue Literary Press.

https://www.autodesk.com/redshift/al-bahr-towers/.

https://thetowerinfo.com/buildings-list/doha-tower/.

March, L., & Steadman, P. (1971). *The geometry of environment*. Cambridge: The MIT Press.

Martin, G. (1982). *Transformation geometry: An introduction to symmetry*. New York: Springer-Verlag.

Maspero, G. (1974). *Manual of Egyptian archaeology and guide to the study of antiquities in Egypt*. New York, NY: Putnams Sons Press.

Park, J.-H. (2018). Subsymmetries for the analysis and design of housing facades. *Nexus Network Journal: Architecture and Mathematics, 20*(1), 251–266.

Park, J.-H. (2000). Subsymmetry analysis of architectural designs: Some examples. *Environment and Planning B: Planning and Design, 27*(1), 121–136.

Sidawi, B. (2012). A conceptual analytic model of the vocabulary of the Islamic architectural heritage. *Emirates Journal for Engineering Research, 17*(1), 47–56.

Architecture, Narrative and Interaction in the Cityscapes of the Assassin's Creed Series

A Preliminary Analysis of the Design of Selected Historical Cities

Danilo Di Mascio

1 Introduction: Architecture and Video Games

This chapter presents a preliminary analysis of some historical cities in selected episodes of the Assassin's Creed series. In particular, the work investigates aspects related to architecture, narrative and interactivity and considers how they intersect with each other within those cityscapes.

The artistic and cultural value of video games is being increasingly recognised, and the successful recent exhibition at the Victoria and Albert Museum in London further demonstrates this. The exhibition, entitled "Videogames: Design/Play/Disrupt" (Foulston and Volsing 2018), was the first to explore and disseminate the design process behind video games. Moreover, Foulston and Volsing (ibid), in the introduction of the publication that accompanied the exhibition at the V&A, stated that they *"[…] looked at fields such as architecture, a subject that grapples with the very large issue of physical scale (how can you exhibit a building within the walls of another?)[…]"* because it is a subject which, together with theatre, asks *"[…] curators to look beyond the finished form. Learning from methods of display and interpretation employed by these areas we sought to define a new curatorial language for videogames: one that enables us to travel beyond the game itself and to make visible its design"* (Foulston and Volsing 2018: 10).

Thanks to technological advancements, video game companies are now able to create virtual environments which have an impressive level of detail and realism, and the quality and size of these environments are likely to increase. The design of virtual environments, called level design, has always played a central role in the creation of a video game, as important as the design of other essential elements such

D. Di Mascio (✉)
Department of Architecture and 3D Design, School of Art, Design & Architecture, The University of Huddersfield, Huddersfield, UK
e-mail: D.DiMascio@hud.ac.uk

J.-H. Lee (ed.), *A New Perspective of Cultural DNA*, KAIST Research Series,
https://doi.org/10.1007/978-981-15-7707-9_9

as the characters. More powerful machines and digital tools allow designers and artists to create highly complex environments which are inspired by, or are fictional recreations of, real architectures and places. In fact, in modern video games, the role of architecture is becoming more and more prominent.

Despite the ever-increasing diffusion of, and interest in, video games by both the general public and the academic environment, the number of scientific studies on the topic remains limited. In architecture and related disciplines, studies dealing with aspects of video games that go beyond simple visualisation are still quite rare. Some examples worth mentioning are represented by Zarzycki (2016), Borries et al. (2007), Totten (2014) and Di Mascio (2010, 2017). However, the video games field can still be considered mostly overlooked by architectural research, and for this reason, it presents a vast number of research opportunities. Furthermore, considering the number and complexity of disciplines which converge in each video game, it is clear that they can be studied and analysed from multiple perspectives.

The Assassin's Creed series is one of those remarkable examples that prove how the design of cities and architectures is becoming increasingly critical in modern video games. However, despite its success, and its richness and complexity in terms of design, art, technology and cultural elements, this series of video games has not been adequately investigated, especially from perspectives close to the architectural field.

Each episode allows the user to experience different historical buildings and places, and the design of each of these is always informed by extensive historical research. Moreover, it can be said that all these games have a similar cultural DNA, in that there are always cultural elements pertaining to different historical periods intertwined together. These elements form portions of a cultural DNA which are present in every episode, similarly to how hereditary characteristics constitute portions of human DNA. Architecture and its related components, from spaces to architectural objects and narrative, form some of these elements together with interactivity and other aspects of these video games. Hence, the same DNA principle can be applied to other historical information.

This chapter presents a preliminary analysis of architecture, narrative and interaction within some cities in three episodes of the franchise. By laying the groundwork for further studies, the research has multiple values in design and education. One of its key aims is to reflect upon the design of fictional cities and places, an aspect relevant to several fields such as architecture, video games, cinema and comic books. A consideration of the design of fictional cities, in this case within video games, can also encourage reflection upon the design of real urban environments.

2 Cities in Video Games: Between Real and Imaginary Places

The variety and number of cities in video games are expanding an already remarkable set of cities described or introduced by other cultural works in fields such as architecture, literature, painting, cinema and comic books. Moreover, cities in video games represent a type of virtual place, alongside the many other spaces which constitute the manifold set of video game worlds (Morris and Hartas 2004).

Compared to those other spaces, 3D cities in video games (which allow users to roam freely and explore wide urban areas as well as the interiors of selected buildings) present architectural and spatial complexities which are close to those that can be found in real-world architecture and places. The fact that these virtual cities can be explored in real time, and by travelling to and from various locations using different paths and at different paces, makes the design and experience of 3D cities in video games much more complex and rich than their counterparts found, for example, in cinema. In a movie, a city is usually described by different predefined framings which can be visually detailed and evocative (e.g. the cities in Star Wars), but, on the other hand, only partial views of those places are provided and there is none of the active engagement through users' direct experience which is available in a virtual city. In a video game's virtual city, the user can freely decide where to go, how to reach a specific location and how much time to employ. He or she can usually stop and admire any details for as much time as he/she desires, in a similar way to in the real world. This kind of interaction is the key aspect that characterises video games and diversifies them from all other artistic media.

Cities in video games can be divided, to some extent, into three main categories: cities that are reproductions of real places (existing or lost, modern or historical); cities which are fictional locations but clearly inspired by real cities (e.g. Grand Theft Auto V's Los Santos); and cities that are completely imaginary (e.g. Dishonored's Dunwall). The percentages of realism and imagination are variable in each of those categories and in each of those individual cities.

Moreover, imaginary cities can also make extensive use of modern and historical architectural vocabularies. The city of Yharnam in the video game Bloodborne (FromSoftware 2015) is a remarkable example of an imaginary city characterised by intricate spaces and evocative glimpses made possible by creative use of Victorian architecture, such as spires, buttresses, columns and sophisticated decorations (Fig. 1) inspired by real cities including London and Edinburgh.

Another excellent example in this regard is the city of Rapture in the video game Bioshock (Irrational Games 2007). For this underwater city, the designers used an architectural vocabulary inspired by Art Deco architecture. Both examples convincingly show how architectural elements pertaining to different historical periods and styles can be used to create entirely imaginary places which demonstrate a strong identity.

Fig. 1 In-game screenshot collected during the exploration of Yharnam in Bloodborne. The imaginary city is characterised by creative use of Victorian Architecture. *Source* Author's personal archive

The cities in the Assassin's Creed series pertain to the category of cities that can be defined, to some extent, as interpretive and creative reconstructions of real historical places.

3 The Assassin's Creed Series and Its Historical Cities

The series of Assassin's Creed started in 2007 with a game of the same name (Ubisoft Montreal 2007). Since then, eleven video games have been published, with the last one, entitled "Assassin's Creed Odyssey", published in 2018. The whole series has been hugely successful and this is confirmed by the number of copies sold, which makes the franchise one of the best selling in the history of video games (Wikipedia Contributors 2020). The games are defined as third-person action adventures.

Games from this famous series have been studied and analysed by several scholars who have focused on various aspects, such as how the virtual experience of a Renaissance Florence (in Assassin's Creed II) which includes anachronistic monuments may influence gamers' perception of the actual city and its landmarks (Dow 2013). Another study has explored the realism of how historical facts are represented, and how the player experiences these through specific characters and their social and cultural background (Shaw 2015). However, there is a lack of research focusing on the intersection between architecture, narrative and interaction within these historical virtual cities.

One of the distinctive characteristics of the series since its first episode (which has also contributed to its success) is the possibility of exploring large historical cityscapes densely populated by Non-Player Characters (NPCs). For the first time, users have the opportunity of experiencing historical situations and locations in a highly immersive and interactive way. The leap from what people can experience by reading historical information in a book, or watching a documentary about a historical period or a movie with a historical setting is huge, as is the emotional engagement. It is possible to state that part of the success of the series is thanks to the use of credible historical settings and events. *"History provides a rich source of artistic references that game designers have been swift to exploit. One advantage of drawing on a historical period for inspiration is that people recognise it"* (Morris and Hartas 2004: 46). This also explains the success of many movies and novels which are part (together with the Assassin's Creed series) of the so-called historical fiction genre.

The connection with heritage is so strong that a compilation which included the first five episodes of the series, sold for the previous console generation (PlayStation 3), was called the "Heritage Collection". Hence, it is evident that the design of these virtual cities and their architecture (as well as the design of the characters and their historical costumes) assumes an essential role within the production of the whole video games series. A precise gameplay choice characterises another essential feature of the series. The character controlled by the player can free roam, run and climb most of the 3D objects within the virtual game world, which mainly comprise buildings located in urban environments.

The storyline of the series is not limited to particular historical periods, however. In fact, all the time travel is explained by means of an interesting sci-fi narrative. All the historical adventures are memories which the protagonist can experience through the use of a sophisticated technological device called an Animus (Miller 2015). Through this virtual reality device, the main character can access genetic memories linked to his ancestors (which represents a clear link with the concept of DNA, even within this video games series).

The first game of the series allowed the designers to develop a methodology for the creation of the virtual historical cities. Hence, it is possible to state that there are elements, especially the way in which cultural references have been studied and used, and the way urban patterns have been generated, which constitute part of this DNA in every episode.

Considering the number and size of the video games of the series, some limitations to the scope of the research are required. In particular, this chapter focuses on cities in three episodes, namely, Assassin's Creed (Ubisoft Montreal 2007), Assassin's Creed Unity (Ubisoft Montreal 2014) and Assassin's Creed Syndicate (Ubisoft Montreal 2015). Moreover, the research is still in progress, and further aspects of these video games will be explored and presented at a future date.

In the research discussed in this chapter, the focus is not limited to representation, but considers architecture, narrative and interaction, including how these elements relate to, and are influenced by, each other. Therefore, this research does not focus on the accuracy of the historical reconstructions of the cityscapes.

4 Methodology: Theory and Case Study

One aspect of the case study of the Assassin's Creed series and its related cities constitutes theory development. In fact, one of the purposes of this research is to further develop and test a theoretical framework for reading and systematising 3D representations of cities in video games, as described in Di Mascio (2017).

The methodology applied during the research is the result of a combination of different approaches and sources which support the analysis of the selected cities. These include the application of a theoretical framework, direct experience of the selected games and the study of several sources related to the video games series. Personal experience in level design for video games, as well as reflections developed and published in other pieces of writing such as Di Mascio (2010), has supported the study, while experience and reflections developed by analysing architecture and cities in other video games have also informed the present research.

The aspects, concepts and definitions of 3D representations of cities identified, developed and described in Di Mascio (2017) have been used as a basis for analysing the selected cities. The analysis has also been supported by concepts presented in publications in the architecture field, such as Kevin Lynch's seminal work, "The Image of the City" (Lynch 1960).

The selected cities were analysed through direct experience of the video games, with the theoretical framework being applied during the virtual explorations. This is similar to the way in which an architect uses his or her background to investigate a real location during a site visit and analysis. As video games are characterised by their interactive components, it is essential to play/experience them directly in order to explore and understand their most essential features, such as the virtual environments and the way users can interact with and experience them. It is not enough to see a recorded gameplay session or to sit next to a person playing a game.

The theoretical framework and virtual explorations have also been supported by the study and analysis of several relevant resources, starting with the official Art Books of the selected chapter of the Assassin's Creed series (Ubisoft 2007; Davies 2014, 2015). In particular, the first Art Book (Ubisoft 2007), which covers the genesis of the first video game of the series, explains the main design choices that have characterised the whole series. In addition to the books, conference papers, websites, journal articles and interviews have also been studied.

5 Reconstructing Historical Cities: A Design Approach

Before starting the analysis of architecture, narrative and interactivity within the historical cities in Assassin's Creed, it is relevant to know how Ubisoft's designers approached the challenge of reconstructing large historical places. It is evident that the challenge may be approached in several ways.

From the first instalment of the series, the game designers opted for an approach that balanced accuracy with creative interpretation and design. For example, the three main historical cities in the first Assassin's Creed episode, Acre, Jerusalem and Damascus, were built upon archival materials such as historical maps. However, in order to make those cities aesthetically more intriguing, and with layouts that better supported the gameplay dynamics, the designers developed and applied an approach of representing and creating the cityscapes in a more stylised way (Ubisoft 2007). The same method has been used for the other cities designed for subsequent episodes of the series.

The designers state that the location of the main landmarks is frequently quite accurate and is based on available historical maps and information including both writing and images. When historical information was lacking, the designers had more freedom to interpret the location and add more personal ideas. An example showing the result of this approach is the design of the Masyaf Fortress for the first episode of the series. The historical information about this fortress was scarce, and hence, the designers had the freedom to also interpret the location from a narrative point of view. Rafael Lacoste, Art Director of the first episode of the series, states that, *"The idea for Masyaf was to make the midway point between the fantastic and the realistic setting.[…] I have tried to exaggerate the steps, the height of the castle so it's more dramatic and very stylized"* (Ubisoft 2007: 107). The same approach has been adopted for elements in other episodes of the franchise. In general, the designers have tried to respect historical information and facts as far as possible, but when there was less certainty and there were gaps in the information, the designers had the freedom to develop those locations creatively.

Thus, while the design of the virtual cities is as historically accurate as possible, it is also stylised in order to support aesthetic, narrative and gameplay mechanics. At the same time, however, the final representation is always believable, in the same way as places and architectures in movies such as Lord of the Rings, Harry Potter or Star Wars are believable. All these are works of fiction, whose main aim is to entertain users.

6 Analysis of Architecture, Narrative and Interactivity and Their Intersection

This section describes architecture, narrative and interactivity in the cityscapes of the three selected episodes of the series and considers how these elements intersect with each other in various ways. The analysis starts with interaction/gameplay because, as stated before, it is the main characteristic which differentiates a video game from a book or a movie.

6.1 Interaction—Gameplay

The way the player interacts with the virtual environment is influenced by the game genre, gameplay/design choices, storytelling and game-world structure. As mentioned in Sect. 3, Assassin's Creed is a third-person action-adventure game, with a relevant stealth component. The player can perform various actions, including walk (Fig. 2—above), jump, climb (Fig. 2—below), talk, fight, drive and swim. Every mission can be completed in more than one way.

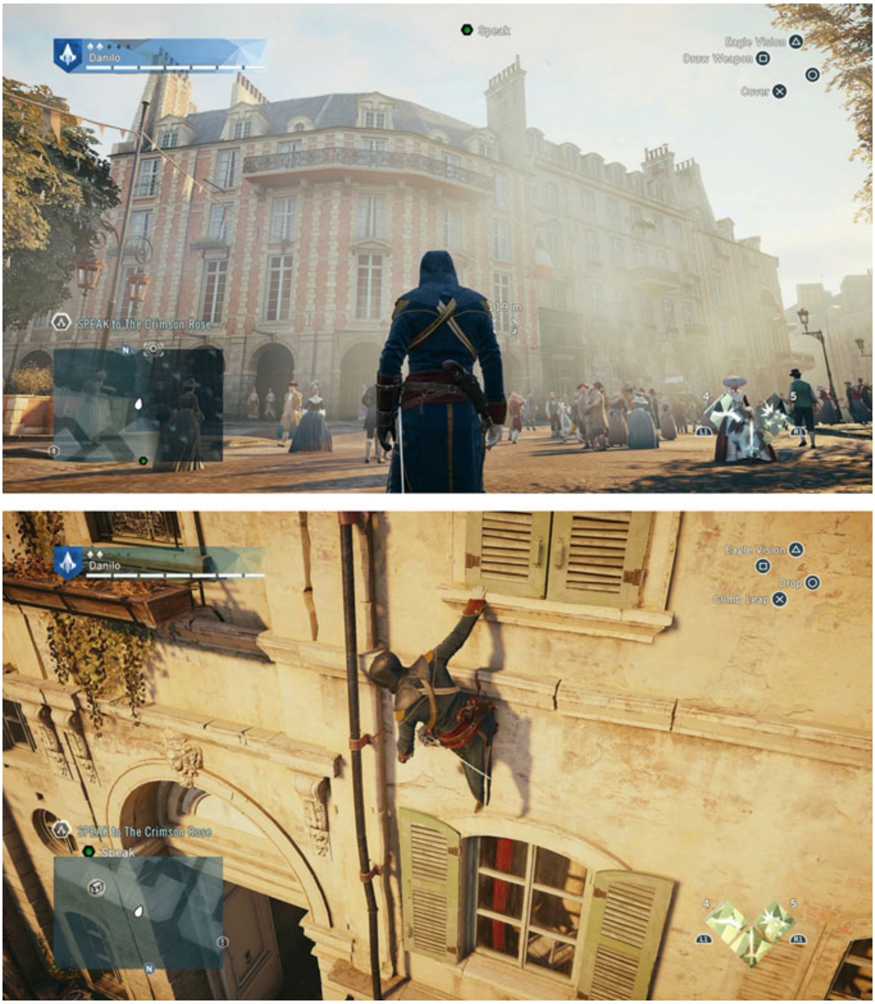


Fig. 2 In-game screenshot collected during the exploration of Paris in Assassin's Creed Unity. The main character is shown walking around the city (above) and climbing a building (below). *Source* Author's personal archive

The game has an open world; this means that the player can free roam around the different cities and areas of the virtual environments without the need to follow a pre-planned path typical of video games that present a so-called linear structure. The gameplay and the storyline influence the design of the game world. Different historical eras also allow different ways of interacting with the game world.

6.2 Narrative Layers

In Assassin's Creed, as in many other video games which present strong narrative components such as Dishonored (Arkane Studios 2012), Bioshock (Irrational Games 2007) and Bloodborne (FromSoftware 2015), narrative permeates several aspects of the series, starting with the game world. As in many other fields, here the word narrative can be interpreted in different ways. In the context of this research, narrative is considered in a twofold way: as a sequence of facts (where characters and their actions are the protagonists), and as meanings, emotions and information communicated by objects, buildings and places (and through their interactions with characters). Considering this latter interpretation, within the games' cityscapes, narrative is embedded in every detail including buildings and streets, and even in small objects and decorations such as lampposts, signs, statues, books and scrolls left in strategic locations.

6.2.1 Main Plot—Storyline

The entire series of Assassin's Creed describes a conflict between two opposing factions, the Assassins and the Templars. Each of these factions envisions and pursues a different way of addressing humankind; the Assassin's vision is characterised by providing each person with complete freedom to shape his/her own life, while the Templars' vision aims at humanity being ruled by an elite group of people (Miller 2015). Each video game of the franchise explores this centuries-old battle during different times in history and in different locations; this is why every episode is contextualised to a different historical setting and period.

6.2.2 Historically Contextualised Plots

In each era, there are different events which take place, and there are various characters (including fictionalised versions of real historical figures such as Dickens, Darwin and Bell in AC Syndicate) who can directly or indirectly influence the main story and the protagonist's missions and adventures. The stories of each episode can be defined as historically contextualised plots because they are intertwined with, or characterised by, the historical events of the specific era in which each chapter takes place. For example, during the first phases of Assassin's Creed Unity, the protagonist, Arno Dorian, is imprisoned in the Bastille and needs to escape together with another prisoner, an Assassin of the Brotherhood. Thus, real and fictional events merge into each other.

6.2.3 Background Setting

Each city in every episode is part of a broader environment that serves as a frame for the main story. In the first Assassin's Creed, the background setting is the Holy Land during the Crusades. The second game of the franchise opted for a different context, the Italian Renaissance. In Unity, the designers selected France, and in particular, Paris, during the French Revolution. In Syndicate, England during the Industrial Revolution represents the main background of the game. Obviously, the background setting influences many aspects, including architecture, dress, weapons, behaviour and characters' motivations.

6.2.4 Main Setting

Each game of the series presents one (sometimes more) main historical setting. In the first game, Ubisoft recreated three main cities, Damascus, Acre and Jerusalem, and a portion of the landscape that connects and includes them. In AC Unity, the primary historical setting is mainly represented by one single big city, Paris in 1789 during the French Revolution, even if it is possible to visit other locations including Versailles. Moreover, thanks to a specific narrative choice (explained as a time anomaly) the player can also explore Paris during the Belle Époque.

Syndicate, the subsequent episode in the franchise, is mainly set in Victorian London during the Industrial Revolution, but before arriving in the English capital, the protagonists explore a few other introductory areas.

6.3 Architecture and Urban Representations

6.3.1 Urban Pattern

The urban pattern of each of the cities is based upon historical data, such as historical maps. Obviously, eighteenth-century Paris and nineteenth-century London are better documented than Damascus, Acre and Jerusalem during the Crusades. The creation of the virtual London, in particular, has benefitted from documents not available for the previous cities, such as extensive historical photographic documentation further expanded by new photos taken first hand (Kerr 2015).

Assassin's Creed Unity shows Paris during the French Revolution (Fig. 3), hence before Haussmann's renovation of the city which introduced the famous boulevards that made the city famous worldwide. In this way, the virtual reconstruction of Paris presents and allows the player to experience many narrow spaces of the mediaeval

Fig. 3 Screenshot of Paris taken from Assassin's Creed Unity. *Source* Author's personal archive

Paris that have been completely lost since the big renovation project. It is Paris at a specific moment in time. To design this virtual Paris, the designers consulted more than 150 maps (Webster 2014).

London in AC Syndicate is represented as a busy city during an epochal technological revolution (Fig. 4) which introduces new means of transportation such as trains, carriages and steamboats. These new means of transport affect interaction and

Fig. 4 Screenshot of London taken from Assassin's Creed Syndicate. *Source* Author's personal archive

gameplay because they allow new ways of experiencing the city. They also influence the urban pattern towards a more realistic design, in order to accommodate citizens walking on the pavements and crossing the street at corners (Kerr 2015). London's main streets are generally wider than the mediaeval streets of Unity's Paris, and they can accommodate usable carriages which allow faster transportation between the areas of the city. Even where some of the Paris streets are wide enough to contain carriages, the turmoil of the revolution makes them unusable, and in fact, some carriages are scattered around the city and abandoned at the edges of the streets.

6.3.2 Architectural Style

The architectural style of each city reflects the real architectural styles present in that specific city at that particular moment in history. Moreover, each district of a city can be characterised by a specific architectural style, because that particular area may have been developed during a specific historical period. Districts are also one of the key elements which constitute the image of a city, as highlighted also by Kevin Lynch in his publication (Lynch 1960).

Each of the six districts in AC Unity presents a distinctive architectural style which is also, to some extent, a representation of the social, cultural and economic conditions of the neighbourhood. For instance, Le Marais district features elegant and decorated buildings, which reflect the luxurious lifestyle of its inhabitants. In contrast, La Bièvre depicts neglected constructions and an urban environment negatively affected by the presence of many tanneries, all elements that reflect the poor condition of its inhabitants. These aspects are further emphasised by the precarious and fragile nature of many wooden structures, which give the impression that they could fall down at any time.

In Syndicate, the different areas of London are also characterised by specific architectural styles and typologies, and the architectural style of each neighbourhood supports a particular narrative and gameplay. The use of specific historical periods also allows the introduction of structures, spaces and materials not available during previous historical periods and hence in previous historical cities of the series. For example, London during the industrial revolution had transformed itself into the first modern metropolis, and this is effectively communicated by the way the architecture of the city has been represented. The buildings generally have more storeys than their Parisian counterparts, which further emphasises the character of the capital of the biggest empire existing at that time. Like Paris, London was a city of contrasts, but these contrasts were even more distinct. Whitechapel is characterised by poverty and irregular, intricate and gloomy back alleys, ideal places to host shelters for criminal gangs, while Westminster shows order and wealth. From above, however, both cities present an expanse of rooftops and chimneys, ideal playgrounds for an acrobatic assassin.

6.3.3 Landmarks

Each city in the video games of the series, like its real historical (and modern) counterpart, features several architectural monuments and buildings which have multiple functions. The role of landmarks in a real city is also highlighted in Kevin Lynch's "The Image of the City", where he describes them as "point references" (Lynch 1960: 78) which are visible from long distances and from various positions around the city.

In Paris during the French Revolution, some of the main landmarks include Notre Dame Cathedral (Fig. 5—above), Sainte-Chapelle, the Sorbonne (Fig. 5—below), the Conciergerie, the Bastille and the Pantheon.

Each landmark has been represented with a high level of detail, and a considerable amount of research material has been used to support each digital reconstruction. In Syndicate's Victorian London, it is possible to find many landmarks including Trafalgar Square, the Houses of Parliament, St. Paul's Cathedral, Waterloo station and St. James' Park. However, in both games, the designers decided to include only a selection of landmarks. Obviously, cities such as Paris and London contain an abundance of relevant monuments, which would have been difficult to recreate due to time and technological constraints. For example, it took two developers two years to create Notre Dame (Miller 2015: 203). Sometimes, specific landmarks are placed far from where the protagonist starts the adventure; this approach creates a sense of anticipation, which increases the desire to reach this unique location.

The function of each landmark is not only symbolic, narrative and aesthetic, but is also connected to the gameplay. At the beginning of each gameplay session, the map of each city is blurred. Hence, the streets and buildings are not easily recognisable. In order to make a portion of the map visible, the character needs to reach the highest point of the area, usually represented by an important monument which serves as a viewpoint, and perform an action called "synchronisation". After being "synchronised" with a specific vantage point, the portion of the map surrounding the landmark becomes visible.

On other occasions, the landmarks are the setting of a mission where the protagonist needs to use a new device. In Syndicate, one of Evie Frie's quests is to help the scientist Alexander Bell by installing some fuse boxes in places located in the higher parts of the Houses of Parliament, including Big Ben. This mission also allows the player to become familiar with the grappling hook/rope launcher, a new device which enables the character to ascend tall buildings and landmarks very quickly (Fig. 6). The nature of the device fits well with the scientific discoveries of the period and appears in a setting that is vaguely influenced by the steampunk genre.

6.4 Atmosphere

Atmosphere is communicated in different ways and is also influenced by some of the elements previously described, such as narrative, architectural style and landmarks. In terms of narrative, for example, the selection of a specific historical period with

Fig. 5 In-game screenshot collected during the exploration of Paris in Assassin's Creed Unity. Observing landmarks from different points of view: Notre Dame Cathedral seen from the ground in the middle of a narrow street (above), and the Pantheon (below) seen from another landmark, the Sorbonne. *Source* Author's personal archive

its related events influences the atmosphere of a place. Other aspects that play an essential role in the creation of atmosphere are lights and colours, often inspired by creative works from other fields, including literature, paintings and movies.

Each of the three cities in the first Assassin's Creed, Damascus (Fig. 7—above), Acre (Fig. 7—below) and Jerusalem, presents a definite atmosphere (Ubisoft 2007). Damascus has been represented as a typical Saracen city, inspired by the famous Arabian stories, "One Thousand and One Nights". The city has a joyful atmosphere,

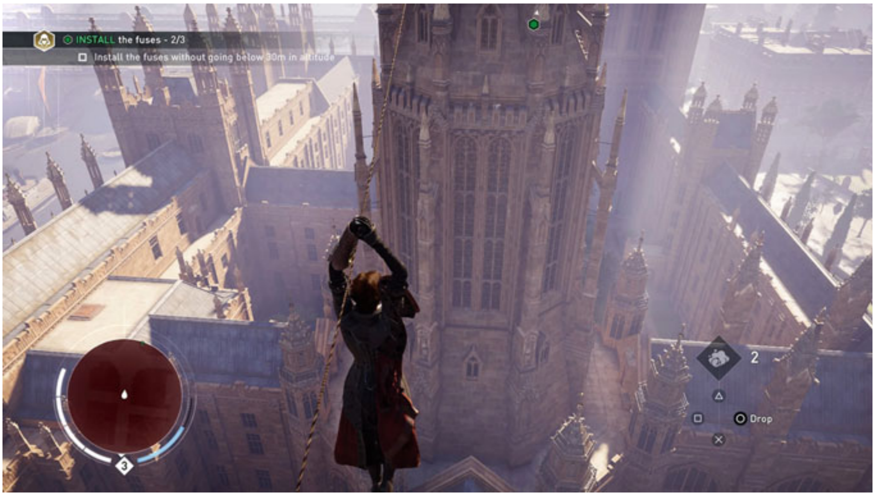


Fig. 6 In-game screenshot collected during the exploration of London. The character is shown using the grappling hook/rope launcher to ascend the Houses of Parliament. *Source* Author's personal archive

highlighted by the use of a yellow filter which communicates a warm feeling. In contrast, Acre is depicted as a colder and darker place, because it has just been a battlefield between two opposing factions, where many executions took place. This gloomy atmosphere is further emphasised by the use of a blue filter, foggy weather and several buildings in ruins. The place seems like a borderline city, where two cultures are mixed, and this is also made evident by the architectural styles of the buildings, which are part European, part Arab. Jerusalem is represented by an atmosphere that suggests a mystical mood, communicated by a green filter (Ubisoft 2007). The three coloured filters help in creating visual cohesion.

In the Paris and London of Unity and Syndicate, the different districts of the cities have also been designed to represent the specific character of each area and its inhabitants. The visual style of Paris has been inspired by the movie "Perfume: The Story of a Murder" (Miller 2015: 203), whilst in Syndicate, the different weather conditions and times of day are more emphasised than in Unity. The official artbook shows a study where the same location has been represented at different times of day and with various weather conditions (Davies 2015). The new industrial era and the introduction of gaslights create a very atmospheric setting during the night and at twilight, clearly inspired by the paintings of John Atkinson Grimshaw (Davies 2015). Sometimes, weather conditions further emphasise the drama of specific missions, for example by reducing visibility during heavy rainfall at night.

More recent episodes (such as Assassin's Creed Origins) emphasise such atmospheric qualities even more, thanks to the changes between day and night through a well-balanced dynamic cycle. This cycle also influences the behaviour of Non-Player Characters (NPCs) and hence the player's actions during specific missions.

Fig. 7 In-game screenshots collected during the exploration of Damascus (above) and Acre (below) in the first Assassin's Creed. The architecture and colours emphasise the contrasting atmospheres of the two cities. *Image source* Author's personal archive

7 Reflections

This analysis of the design and representation of selected historical cities in the Assassin's Creed series has identified several factors of interest and value. The following reflections are grounded in this research but generalise concepts and ideas that can be applied to wider contexts.

First, it has been revealed how historical information has been interpreted and blended with creative design work in order to create historically believable, aesthetically evocative and engaging virtual cityscapes. Because of the amount of research and the complexity of the design process behind the creation of these virtual cityscapes, analysis of the whole process has strong educational value. People outside the video games field may find it challenging to understand, and hence to appreciate, the amount of knowledge, research and work that goes into the design of just one element presented in this series. Hence, this work may help to make people more aware of the artistic and cultural qualities of video games and enable them to understand and appreciate relevant aspects of the design of what can be considered one of the most relevant and innovative cultural/artistic expressions of the present time.

Second, the analysis has highlighted aspects related to architecture, narrative and interaction, as well as how these elements intersect each other. A landmark such as Notre Dame or the Houses of Parliament can represent an important cultural element and communicate interesting historical information; it can also be the scene of a crucial mission/narrative event and the background for other explorations.

Another significant value of this research is that it allows the elaboration of ideas and reflections about the design of virtual environments, fictional places and cities based on historical data. This aspect will become increasingly critical with the broader diffusion of virtual reality devices. It is relevant to highlight that each of the cities in the Assassin's Creed series represents an interpretation of a real counterpart. This means, for example, that Unity's Paris is not the definitive and unique version of the city during that specific historical period. Even if the design of these fictional cities is based on historical research, *"[…] that may not be as simple as it seems. There is no single middle ages, for instance. Rather, there are many visions and many shades of subjectivity in how to mine and work material from such a rich vein"* (Morris and Hartas 2004: 46). The concept that every city can be interpreted in several ways is also clearly demonstrated by works in other fields, such as cinema or paintings. In the publication "A Capital View. The Art of Edinburgh" (Popiel 2014), it is possible to admire 100 artworks that depict the Scottish capital. Each of those scenes uses a different point of view and graphic style and focuses on various subjects and moments in Edinburgh's rich history. Each representational technique supports the narrative that each artist desired to represent and communicate. For this reason, there can be infinite interpretations of Paris and London. For the Assassin's Creed series, the primary aim is to create an entertainment product, not a historically accurate research document. However, even if the series is not intended to teach history, it can trigger people's curiosity, and hence an interest in knowing more about specific historical periods.

Finally, as mentioned in Di Mascio (2017), the study of design and representation of cities in video games can encourage reflection upon the design of real cities. Many cities around the world are thought to "look all the same" (Greenfield 2016), and many new interventions lack identity and character. The analysis of cities in video games can trigger reflections about architecture and narrative, perhaps prompting a consideration of how to design places that are not just functional but also enjoyable and meaningful. This is why some people just enjoy exploring these virtual cities, sometimes even more than undertaking the missions to advance the story (Webster 2014). It is essential to have a final vision shaped by research and concept design (Di Mascio 2017) and to consider the human experience of a building or place as one of the most critical variables. These virtual environments, which present complex and fascinating cityscapes, can also trigger a sense of wonder, discovery and curiosity that can inspire new interventions in the real world. Kevin Lynch shared the same ideas by saying that *"[…] we need an environment which is not simply well organised, but poetic and symbolic as well"* (Lynch 1960).

8 Conclusions and Future Developments

In this chapter, a preliminary analysis of how architecture, narrative and interactivity intersect with each other in the design of some cities in selected episodes of Assassin's Creed has been presented. The Assassin's Creed series, because of the number of episodes, the rich and varied historical settings, and the quality and complexity of the historical reconstructions, may be one of the most studied series of video games. However, the number of studies of these video games from an architectural viewpoint view is still quite limited.

Each city in the Assassin's Creed series presents a distinct character linked to its architecture, atmosphere, narrative and landmarks, all influenced by the specific historical and cultural context in which each city is set. As in the exploration of real cities, the virtual investigation of virtual cities can trigger further research questions which have not previously been formulated. Moreover, such analysis of the design of cities and virtual architecture in general may encourage architects and urban designers to reflect upon the design of real cities.

One of the future aims is to continue analysis of the selected cities and expand the number of case studies to include other cities/locations from the same episodes and further episodes of the series. The same approach will also be applied to cities in other video games. This research is part of a larger research project which aims to investigate concepts and tools in the video games field that can be transferred to the architecture field. A further aim is to visually translate and communicate some of the concepts previously described through diagrams.

References

Arkane Studios. (2012). *Dishonored* [Blu-Ray] PlayStation 4. Rockville, USA: Bethesda Softworks.
Borries, F. V., Walz, S. P., Böttger, M., Davidson, D., Kelley, H., & Kücklich, J. (2007). *Space Time Play: Computer Games, Architecture and Urbanism: The Next Level*. Basel, Boston: Birkhauser.
Davies, P. (2014). *The art of Assassin's creed unity*. London: Titan Books.
Davies, P. (2015). *The art of Assassin's creed syndicate*. London: Titan Books.

Di Mascio, D. (2010). Learning from videogames level design: An educational point of view for architecture. In *CONVR2010 (Proceedings of the 10th International Conference on Construction Applications of Virtual Reality)* (pp. 199–208), Sendai, Japan, November 2010, 4–5.

Di Mascio, D. (2017). 3D Representations of cities in video games as designed outcomes. A critical overview and systematization, in: Janssen, P., Loh, P., Raonic, A., & Schnabel, M. A. (eds.) *Proceedings of the 22nd CAADRIA (Computer Aided Architectural Design Research in Asia) Conference,* Suzhou, China, 5–8 April 2017, pp. 33–42.

Dow, D. N. (2013). Historical veneers: Anachronism, simulation, and art history in Assassin's Creed II. In M. W. Kapell & A. B. R. Elliott (Eds.), *Playing with the past: Digital games and the simulation of history* (pp. 215–232). New York, NY: Bloomsbury.

Foulston, M., & Volsing, K. (2018). *Videogames: Design/play/disrupt*. London: V&A Publishing.

FromSoftware. (2015). *Bloodborne* [Blu-Ray] PlayStation 4. Tokyo, Japan: Sony Computer Entertainment.

Greenfield, A. (2016). Where are the world's newest cities…and why do they all look the same? *The Guardian,* January 28 (online). Available at: https://www.theguardian.com/cities/2016/jan/28/where-world-newest-cities-look-same. Accessed February 1, 2016.

Irrational Games. (2007). *Bioshock* [Blu-Ray] PlayStation 4. Novato, United States: 2K Games.

Kerr, C. (2015). The world design of Assassin's Creed: Syndicate, *Gamasutra* (online). Available at: https://www.gamasutra.com/view/news/253678/The_world_design_of_Assassins_Creed_Syndicate.php. Accessed January 7, 2020.

Lynch, K. (1960). *The image of the city*. London, Cambridge, MA: M.I.T. Press.

Miller, M. (2015). *Assassin's creed. The complete visual history*. London: Titan Books.

Morris, D., & Hartas, L. (2004). *The Art of Game Worlds.* Lewes, United Kingdom: Ilex Press.

Popiel, A. J. (2014). *A capital view. The art of Edinburgh.* Edinburgh: Birlinn Ltd.

Shaw, A. (2015). The Tyranny of realism: Historical accuracy and politics of representation in Assassin's Creed III. *The Journal of the Canadian Game Studies Association, 9*(14), 4–24. Available at: https://journals.sfu.ca/loading/index.php/loading/article/view/157/185. Accessed January 11, 2020.

Totten, C. W. (2014). *An architectural approach to level design.* Boca Raton: CRC Press, Taylor and Francis Group.

Ubisoft Montreal. (2007). *Assassin's Creed* [Blu-Ray] PlayStation 3. Montreal, Canada: Ubisoft.

Ubisoft Montreal. (2014). *Assassin's Creed Unity* [Blu-Ray] PlayStation 4. Montreal: Ubisoft.

Ubisoft Montreal. (2015). *Assassin's creed syndicate* [Blu-Ray] PlayStation 4. Montreal: Ubisoft.

Ubisoft. (2007). *Assassin's creed limited edition art book.* Roseville, CA: Prima Games.

Webster, A. (2014). Building a better Paris in Assassin's Creed Unity. Historical accuracy meets game design, *The Verge* (online). Available at: https://www.theverge.com/2014/10/31/7132587/assassins-creed-unity-paris. Accessed January 13, 2020.

Wikipedia Contributors. (2020). List of best-selling video game franchises. In *Wikipedia, The Free Encyclopedia* (online). Available at: https://en.wikipedia.org/w/index.php?title=List_of_best-selling_video_game_franchises&oldid=934602346. Accessed January 11, 2020.

Zarzycki, A. (2016). Epic video games: Narrative spaces and engaged lives. *Journal of Architectural Computing, 14*(3), 201–211.