Ubiquity, an ACM publication
May 2020

Machine Learning and Computational Design

by Silvio Carta

Editor's Introduction

The use of computers in design is substantially different today from what it was only 30 years ago, and light-years ahead of how things were designed before computers entered the scene 60 years ago. This article discusses the use of computers, more specifically computational design, as a useful tool for designers. Herein, computational design refers to the application of computational tools to design practice.

Association for Computing Machinery
Advancing Computing as a Science & Profession

Ubiquity, an ACM publication
May 2020

Machine Learning and Computational Design

by Silvio Carta

The use of computers in design is substantially different today from what it was only 30 years ago. And light-years ahead of how things were designed before computers entered the scene only about 60 years ago. This article discusses the use of computers, more specifically computational design, as a useful tool for designers (computational design here refers to the application of computational tools to design practice).

Design practice often involves long hours devoted to repetitive tasks such as research, testing, and drawing many options in order to work out the best solution for a given problem. This is particularly the case for architecture and construction. For example, if designers working on a new residential building want to find out the optimal slant for each façade panel to maximize solar gain (amount of sunlight entering through each window), they will need to test several strategies, evaluate them, create models, and simulate results in order to compare the efficiency of each option. Once the designers have found the right strategy, they will still need to revisit each individual panel to evaluate the best angle for performing the task. In this way, a single design job could take weeks of testing, adjustments, meetings with consultants, and could easily lead to frustration with the complexity of the entire process.

For centuries designers accepted this repetitive and often vexing process, largely because they had no choice. It was either this or not design anything of any interest. For example, Renaissance architects created multiple physical models/maquettes to convince their clients of the aesthetic qualities of their projects. It is therefore not surprising that the automation and optimization of tasks appeals to designers and others involved in the design process. This is particularly evident in the case of computational design, whereby computers and software are used as a fundamental part of the process.

Evolution of Computational Tools

During the 1990s and 2000s, designers started to recognize the benefits of using computers to simplify laborious or complex tasks, to save time and resources, and to acquire a higher level of precision and control over the design process. Notably, architecture firm Gehry and Partners made early use of the parametric software Catia to assist the design and fabrication of the Guggenheim Museum in Bilbao, Spain. Around the same time, in 1993, Jon Hirschtick developed Solidworks a CAD (computer-aided design) software that is now used by millions of designers and engineers in product design. The use of CAD, whereby designers use software to replicate hand-drafting more efficiently and accurately, quickly became popular.

After this initial "digital phase," which focused largely on the replication of human tasks by computers, a new way of using computers for design emerged. Recently, a new generation of designers has started including the use of algorithms and computational logic in their work. This approach necessitates a much greater understanding of how computers work and involves the use of computational thinking as a fundamental part of the design process. This new digital age in design includes an awareness of algorithmic logic, datasets, and statistic models. In this respect, as design becomes increasingly data-driven, designers find themselves learning more about this data and developing more effective ways of handling them. The technique of form-finding is a clear example of such an approach, where the shape of a building is not created by the designer but by a combination of algorithms. The designers develop a series of tasks for the computer to perform, they set certain conditions, and then they use computers to run a series of simulations/tests that will eventually return the desired shape. Such approaches have been applied to many fields in design including jewelry, product, and furniture design (Philippe Morel's algorithmic chair is a good example of this); fashion (e.g. 3-D printed garments); graphic design (using software like Casey Reas and Ben Fry's Processing); as well as architecture and construction.

In general terms, design practice is a vast and complex discipline and it would not be possible to automate or optimize all aspects of it. For example, aspects pertaining to intuition, synthesis, and creativity within the design process are hardwired into human nature and cannot be easily replicated by algorithms. However, areas that involve the use of data could be processed by computers and automated in order to augment the design practice. In this sense, computational design should not be considered a substitute for design in general, where automation completely takes over the creative process, it should rather be considered as an additional tool for designers that can, and indeed should, be used to simplify, improve, and extend their work. Through computation, designers can perform quicker, more accurate, and more comprehensive tasks to test concepts and ideas.

Machine Learning

One of the most popular and increasingly used computational approaches to design is machine learning (ML). This is where designers and data scientists work together to generate workflows—a combined series of different steps in a process that result in optimized shapes, spatial configurations, and more performant objects. It is not difficult to imagine how designers frustrated with the angle calculations of multiple facade panels may welcome ML as a very useful tool.

There are three main types of applications where ML is proving particularly beneficial within design processes, specifically in the architecture, engineering, and construction (AEC) industry.

The first of these are analytical tools, where designers use ML techniques to simulate and monitor possible design scenarios. This includes the analysis of existing buildings and public spaces, as well as hypothetical studies, where different factors are tested and building performances evaluated. A recent example of this approach is the MIT SenseAble City Lab's AI Station project which analyzed Wi-Fi signals to understand how passengers move through two train stations in Paris.[1] They used a multi-layered analytical process called deep convolutional neural network (DCNN) to evaluate indoor legibility in the Gare de Lyon and Gare St. Lazare. Indoor legibility is the extent to which a space is organized in a clear and coherent pattern and can be recognized by users. Researchers in this project used photographic images as an input in order to observe people's behaviors and space utilization, as well as visual portions of the spaces.

The second type are design tools that have been developed to support designers in their projects and research, mainly running on open platforms. These include Dynamo (an open-source graphical programming tool), Autodesk Revit (one of the main pieces of building information modeling software used widely by architects, mechanical engineers, and contractors), and McNeel Rhinoceros' Grasshopper (a visual programming language and 3-D modeling software). This group includes applications like Dodo, Owl, or Lunchbox where traditional parametric 3-D modeling programs can be augmented by libraries that add machine learning capabilities (e.g. artificial neural network, nonlinear regression, K-Means clustering, etc.) to be used in conjunction with spatial data modeling.

[1] The project is explained in detail here: https://bit.ly/2TBtmr6.

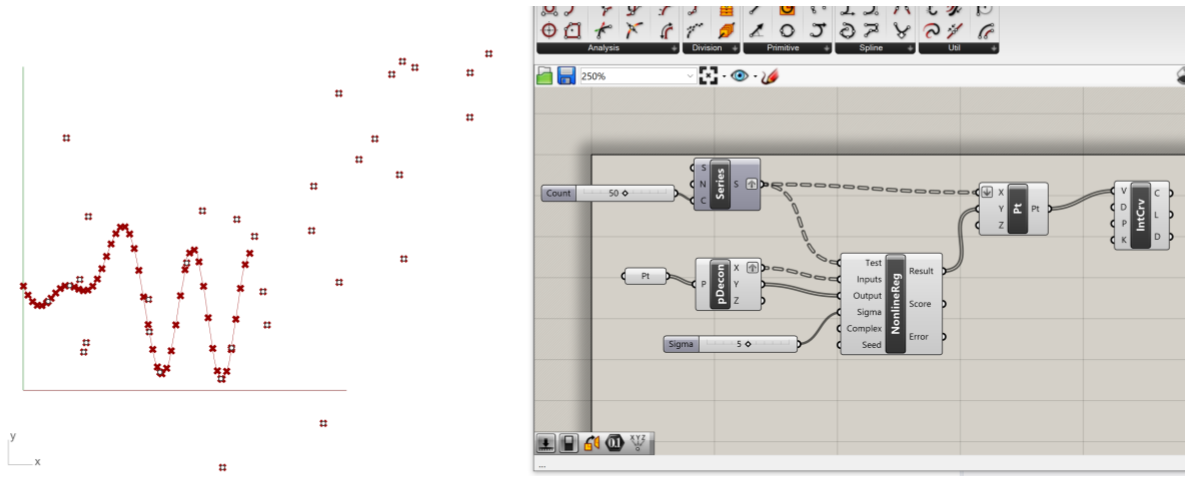

Figure 1. Example of Rhino/Grasshopper interface (Lunchbox library). The example shows a non-linear regression applied to a random series of points. The algorithm predicts the Y-value based on the X-value from a training set. (Image by author.)

The third group includes management and information tools and can be considered as an extended version of more traditional building information Modeling (BIM) systems. These tools are generally referred as part of "City Information Systems" (CIM) and are characterized by a wider application of ML to urban policy and management. New ML-led approaches are being developed across the private and public sectors to combine existing urban information (property, location data, and ordinance survey.) with information generated by the actors involved in the planning process. A particularly successful example is PlanTech, an initiative developed and supported by Connected Places Catapult's digital planning group. The aim of this project is to foster new ways of managing the public digital infrastructure of planning through increasingly more interconnected databases being used by the different actors involved in the planning process, and more automated and optimized services for final users.

ML and Optimization for Design: A case study

One particular example may offer more clarity on how ML approaches are being used within the AEC industry. In a special issue of the *International Journal of Architectural Computing* (*IJAC*), dedicated to the topic of "Intelligent and Informed," Tarabishy and colleagues presented a ML-based model for effectively computing the spatial and visual connectivity potential of a given space. These are important metrics for developing interior layouts, but calculating them in real time can be difficult.

Designers at the global architectural firm Foster + Partners have been working on finding new ways to analyze the spatial configuration of complex building projects (for example large office blocks) at the concept stage of the design process. Spatial configuration is considered in terms of traverse-ability (a measure of how easily users can walk through a space), proximity (the distance between various elements within a space), and visual connectivity (how easily users can see various key parts of the office). The original paper by Tarabishy et al. [1],upon which the following account is based, offers a very good example of how contemporary designers and data scientists can work together to optimize their design outcomes.

Traditionally, Dijkstra's algorithm has been used to calculate spatial connectivity and visibility graph analysis (VGA) to calculate visual connectivity. Dijkstra's algorithm calculates the shortest paths between nodes in a graph, which may represent, for example, various elements in the office, and VGA analyzes what people can see in a given space. The problem with these techniques, especially when applied to architecture, is that their computation can be quite heavy and time-consuming making it difficult to provide a real-time response where it is most needed, for example in the sketch/concept design phase. In their study, the authors explore the use of ML-based techniques to generate surrogate models that substitute/augment these computationally heavy simulations using deep neural networks. These approaches achieve a significant reduction of the computation time along with an optimization of the resources required [1].

In the paper, Tarabishy et al. begin by explaining how the spatial and visual connectivity for a given floor plan can be calculated using VGA and Dijkstra's algorithm, a lengthy process requiring significant computational resources including hours/weeks of calculations depending on the complexity of the floor plan and the availability of resources.

In order to prepare the spatial configuration of a building for simulation and analysis in this way, floor plans need to be reduced to a spatial grid (and parametrized) that includes the key features of the building such as walls, doors, passages, furniture, etc. In this particular study each cell is 0.3 meters and is represented as a graph node for the purposes of analysis. Adjacency is calculated as immediate connection with neighboring cells for the spatial connectivity (excluding unavailable cells like those of walls), and with the rule of "two nodes are connected to each other if you can draw a line without crossing an obstacle for visual connectivity [1]." Tarabishy et al. used Dijkstra's algorithm to calculate the shortest path within the graph (i.e. traversing the graph), while the values of the connectivities were calculated using an isovist graph model.

Recognizing the computational intensity of these simulations, they set about trying to improve the calculation of spatial and visual connectivity by using machine learning. They considered this task in terms of a supervised learning problem, using floor plans as images, and approaching the problem in terms of image processing (rather than semantically, as before). This can be thought of as a mapping exercise between an image of a floor plan (used as an input, with key spatial features such as walls and furniture) and an image of an analyzed plan (used as an output). The output image has some of its pixels unchanged, namely those representing walls and furniture, and others with new values assigned according to the analysis. These values are represented using color gradients.

Having re-expressed the problem in terms of image processing, Tarabishy et al. employed convolutional neural networks (CNNs) to optimize the analysis of spatial and visual connectivity. This method is supported by recent experiments indicating that these CNNs-based algorithms can perform better than others in object detection performance and image classification. In order to be useful for the ML experiment, Tarabishy et al. needed to prepare a set of training data in a suitable format. This data consisted of a large number of different floor plans in raster format and with enough resolution to be effectively processed without unnecessary noise. The researchers were then able to carry out a synthetic data generation using an automated system through a CAD framework (Rhinoceros and Grasshopper). This parametric model allowed them to generate 6,000 bidimensional plans with a variety of spatial configurations (walls and furniture arrangement) to be used for initial testing [1]. The generated images had a resolution of 100 x 100 pixels (each pixel representing 1 meter of physical office space). This was considered a good compromise between the indication of key elements in the plan (expressed in binary terms, with black pixels representing walls and un-traversable elements and white pixel for walkable spaces in the grid) and a reasonable analysis time (which grows exponentially with the resolution).

In order to improve the analysis of these plans with regards to boundaries and the position of users, they introduced a signed distance function (SDF) in all generated plans. This function is used to determine the distance of any point x from the closest other fixed point in a set Ω (indicating the office boundary walls). There is evidence to support that the inclusion of the SDF, along with a binary system of spatial representation, and in conjunction with CNNs can improve the computability of a model for real-time analyses.

Once the dataset is ready for inputting there are a number of parameters that need to be considered and tested before starting the actual training. In this project, the learning rate and the choice of the algorithm were among the most important of these. Tarabishy et al. tested a number of approaches from the U-Net model, which is a type of CNN based on a fully convolutional network (FCN) that has been developed for biomedical image segmentation and that, compared with the original FCN, outputs more precise segmentations with a smaller number of training images including stochastic gradient descent (SGD), Adam, RMSProp, and Adadelta. Among all these optimizers, SGD and Adadelta performed better, with a rapid convergence (correctly mapping black pixels in input with black pixels in output). Eventually they selected Adadelta to run the experiment on the basis that it presented the fastest convergence rate i.e. correctly mapping black pixels in input with black pixels in output than the other options [1].

Machines learn by means of a loss function, a method of evaluating how well the algorithm models the given data. Tarabishy et al. opted to use a combination of the mean squared error

(MSE) and gradient difference loss (GDL) (to introduce a weighted sum) to define the loss function in this case. The latter approach is used in ML, as well as in neural networks, to estimate the performance of a certain model in the optimization process.

In order to obtain the intended level of accuracy , the researchers introduced a generative adversarial networks (GAN) approach based upon two models competing with each other to complete a given task; one generating images that are accurate enough to convince the other model, and a discriminator assessing the outputted images. GAN then converts the loss function into a parameter that can be used to train the model. According to the researchers, "This architecture avoids hand-engineering of the loss function and incentivizes the network to produce images which could be undistinguishable from reality" [1].

They implemented GAN and carried out this training with the Pix2Pix architecture. This is where a network maps input to output images, while at the same time learning a loss function to train the mapping that allowed them to translate an input image to a specific output (instead of a random image), to become a conditional generative adversarial network (cGAN) [1].

They found that [by using] "the Pix2Pix architecture, inference (predicting the output given an input image) for one image is computed in 0.08 s and for the U-Net in 0.032 s for each of the analyses, compared to 15 s for running the actual spatial connectivity analysis and 128 s for running the visual connectivity." The results clearly demonstrate how deep learning surrogate models (and more specifically convolutional neural networks) can significantly reduce the calculation time for an analysis of spatial configuration (0.032 seconds versus 15 and 128 seconds of the methods based on graphs and using VGA and the Dijkstra's algorithm).

Computational Design Today and Tomorrow

This study is fairly representative of the work that progressively hybrid profiles of architects, planners, and data scientists are conducting under the umbrella of computational design. Increasingly, global architecture and urban design firms are establishing in-house research clusters to carry out advanced research in data analysis and visualization, building optimization, simulation, and building performance. Zaha Hadid's Code, KPFui Urban Interface, and Foster + Partners' Applied Research and Development group are all well-known examples of this trend. Research like the study conducted by Tarabishy and colleagues is increasingly relevant both for designers and, more importantly, for everyone involved in the planning, management, and use of cities and public spaces. Research into optimization within design is helping to produce quicker and more accurate simulations, tests, and prototypes in projects where each decision in the design phase corresponds to a large number of actions, costs, months, and years of work and resources. Simulations and analyses of buildings and cities are becoming increasingly more precise, leaving a smaller margin for error and human mistakes. Automation and optimization of processes in design yield better outcomes (that are more performant, functional, and appreciated by users).

There are many aspects of design that are not computational and still rely on human perception, taste, preference, and intuition. However, the systems for the computational aspects have come a long way and will allow for much more complex systems in the future.

A number of challenges still exist for the years ahead. Such approaches are still generally sporadic and characteristic of only a small number of cutting-edge research groups within traditional design firms and universities. In other words, research in optimization, classification, sorting, and machine learning more generally are only possible today within those practices and institutions that can allow investment, in terms of time and resources, into computational research. This tends to occur on a centralized level (research centers, universities, and large design consultancy companies), and is much more difficult (and rare) for small-medium design practices, start-ups, and individuals to engage in. If this line of research is considered to be vital for the progress of design, we are still quite far from reaching a critical mass whereby computational design becomes a collective effort, shared by the entire global design community instead of being promoted by a few small groups of excellence.

There may be a silver lining though. As technology progresses at a fast pace, the AEC industry is constantly pressured to embrace new ways for processes to be automated and optimized, projects to be planned and controlled with higher accuracy, and new data to be produced around each design process (from the exact quantity of certain building materials present in a construction site to metrics to monitor the user's experience in cities). The next 5-10 years will be characterized by an increase in attention to computational design, optimization, and ML techniques to support design. Cities are likely to be increasingly governed by intelligent systems where ML will be a fundamental component, and young designers currently enrolled in a growing number of new university programs that include computational design in their curricula, will be able to contribute more significantly to urban projects. They will consider ML as one of many options in their design toolbox, therefore normalizing and extending the use of ML within the design process. This would allow the extension of computational design and ML to a larger platform, whereby the number of designers engaging in the use of (and proportionally in the research associated with) machine learning reach a critical mass extending to small groups and individuals as well.

Acknowledgements

The central part of this article is dedicated to the description of a project carried out by Sherif Tarabishy, Stamatios Psarras, Marcin Kosicki, and Martha Tsigkari, first published in 2019 in the *International Journal of Architectural Computing* (IJAC). I would like to thank Sherif Tarabishy at the Applied Research and Development group, Foster + Partners, London, UK for his support with this article.

I would also like to thank the editors at *Ubiquity Magazine* for their useful comments and suggestions, in particular Alessio Malizia who supported me with fruitful discussions and good advice. Finally, I would like to thank Anouska Plaut for her help with the overall structure of the article.

References

[1] Tarabishy, S., Psarras, S., Kosicki, M., and Tsigkari, M. Deep learning surrogate models for spatial and visual connectivity. *International Journal of Architectural Computing* 18, 1 (2020).

About the Author

Silvio Carta is an ARB RIBA architect and Head of Art and Design at the University of Hertfordshire, where he is also Director of the Professional Doctorate in Design (DDES). His main fields of interest are digital design, data-driven design and computational design. Silvio is the head of the editorial board of Seoul-based C3 magazine and editor of *Architecture*_MPS (UCL Press). He is currently working on the Machine Learning and the City Reader for Wiley. His last research monograph is “Big Data, Code and the Discrete City. Shaping Public Realms” (Routledge 2019).

DOI: 10.1145/3401842