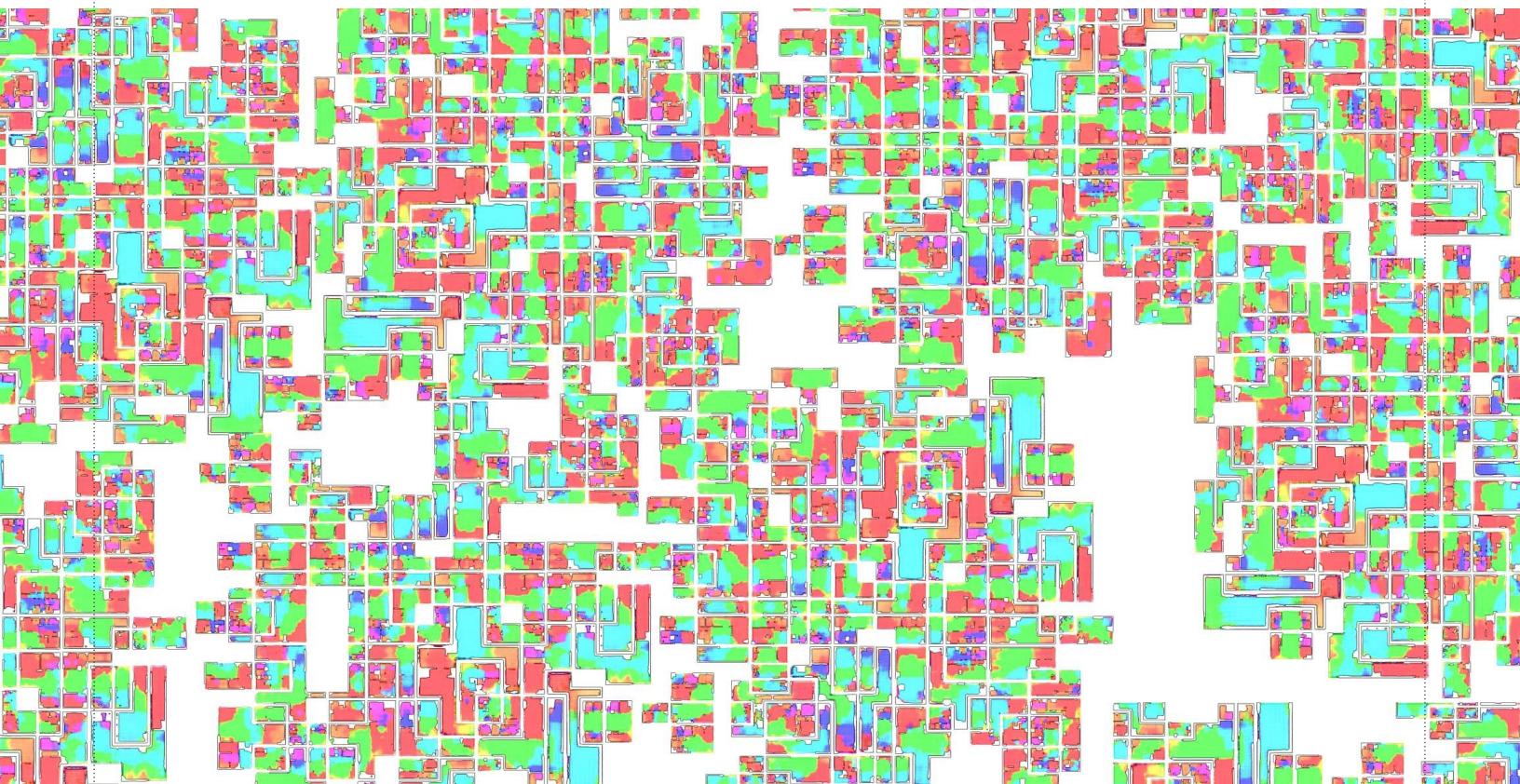


AI & Architecture

An Experimental Perspective

Stanislas Chaillou, Harvard Graduate School of Design | Feb. 24th, 2019



GAN-Generated Masterplan | Source: Author

In this article, we release a part of our thesis, developed at Harvard, and submitted in May 2019. This piece is one building block of a larger body of work, investigating AI's inception in Architecture, its historical background, and its potential for space organization & style.

Artificial Intelligence, as a discipline, has already been permeating countless fields, bringing means and methods to previously unresolved challenges, across industries. **The advent of AI in Architecture, described in a previous article, is still in its early days but offers promising results.** More than a mere opportunity, such potential represents for us a major step ahead, about to reshape the architectural discipline.

Our work proposes to evidence this promise when applied to the built environment. Specifically, we offer to apply AI to floor plans analysis and generation. Our ultimate goal is three-fold: **(1) to generate floor plans** i.e. optimize the generation of a large and highly diverse quantity of floor plan designs, **(2) to qualify floor plans** i.e. offer a proper classification methodology **(3) to allow users to “browse”** through generated design options.

Our methodology follows two main intuitions **(1) the creation of building plans is a non-trivial technical challenge, although encompassing standard optimization technics, and (2) the design of space is a sequential process, requiring successive design steps across different scales (urban scale, building scale, unit scale).**

Then, in order to harness these two realities, we have chosen nested Generative Adversarial Neural Networks or GANs. Such models enable us to capture more complexity across encountered floor plans and to break down the complexity by tackling problems through successive steps. Each step corresponding to a given model, specifically trained for this particular task, the process can eventually evidence the possible back and forth between humans and machines.

Plans are indeed a high-dimensional problem, at the crossroad of quantifiable technics, and more qualitative properties. The study of architectural precedent remains too often a hazardous process, that negates the richness of the number of existing resources while lacking in analytical rigor. Our methodology, inspired by current Data Science methodologies, aims at qualifying floor plans. Through the creation of 6 metrics, we propose a framework that captures architecturally relevant parameters of floor plans. On one hand, **Footprint Shape, Orientation, Thickness & Texture** are three metrics capturing the

essence of a given floor plan's *style*. On the other hand, **Program**, **Connectivity**, and **Circulation** are meant to depict the essence of any floor plan *organization*.

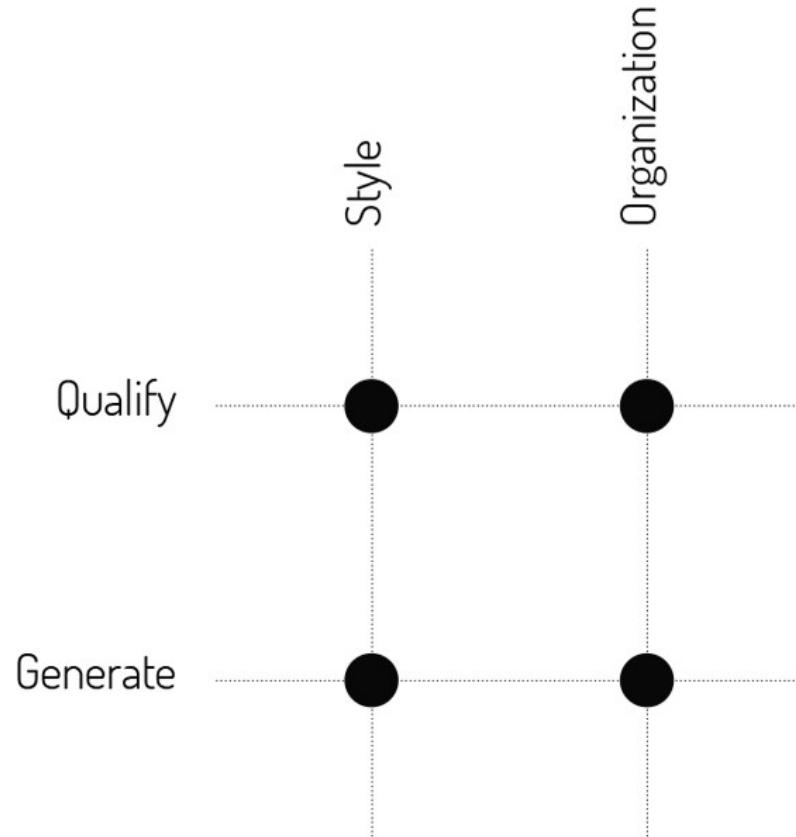
In a nutshell, the machine, once the extension of our pencil, can today be leveraged to map architectural knowledge, and trained to assist us in creating viable design options.

... .

I. Framework

Our work finds itself at the intersection of **Architecture** and **Artificial Intelligence**. The former is the topic, the latter the method. Both have been simplified into clear & actionable categories.

Architecture is here understood as the intersection between *Style* and *Organization*. On one hand, we consider *buildings as vectors of a cultural significance*, that express through their geometry, taxonomy, typology, and decoration a certain style. Baroque, Roman, Gothic, Modern, Contemporary: as many architectural styles that can be found through a careful study of floor plans. On the other hand, *buildings are the product of engineering and science*, answering to strict frameworks and rules -building codes, ergonomics, energetic efficiency, egress, program, etc—that can be found as we read a floor plan. This organizational imperative will complete our definition of Architecture and drive our investigation.



Framework Matrix | Source: Author

Artificial Intelligence will be employed, using two of its main fields of investigation—*Analytics* and *Generative Adversarial Networks*—as an investigative tool.

At first, we will dive into the topic of *Generation*. Using GANs, we offer to educate our own AI systems to architectural design. We postulate that its utilization can enhance the practice of the architectural discipline. This field is as recent as it is experimental and yields to this day surprising results. Our hope is to be able to train it to draw actual building floor plans.

Then, we will come up with a robust analytical framework to qualify and classify the generated floor plans. Ultimately, our goal is to organize the results of our GANs, to offer the possibility for the user to browse seamlessly through the variety of created design options. To that end, the quantity and ubiquity of tools offered by Data Science will prove to be valuable to our investigation.

Through this dual lens, at the crossroad of Style & Organization, Qualification & Generation, we lay down a framework that organizes the encounter of Architecture & AI.

II. Generate

The design of architectural floor plans is at the core of the architecture practice. Its mastery stands as the gold standard of the discipline. It is an exercise that practitioners have overtime relentlessly tried to improve through technology. In this first part, we dive into the potential of AI applied to floor plan generation, as a mean to push the envelope even further.

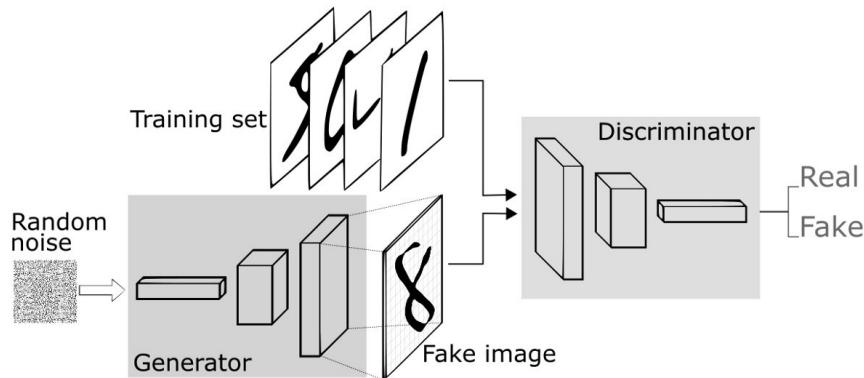
Using our framework, to tackle floor plans' style and organization, we lay down in the following chapter the potential of AI-enabled space planning. Our objective is to offer a set of reliable and robust tools to both evidence the potential of our approach and to test our assumptions.

The challenge is here threefold: (1) choosing the right toolset, (2) isolating the right phenomena to be shown to the machine and (3) ensuring that the machine “learns” properly.

AI & Generative Adversarial Neural Networks

Generative Adversarial Neural Networks—or *GANs*- are here our weapon of choice. Within the field of AI, Neural Networks stands as a key field of investigation. The creative ability of such models has been recently evidenced, through the advent of Generative Adversarial Neural Networks. As any machine-learning model, GANs learn statistically significant phenomena among data presented to them. Their structure, however, represents a breakthrough: made of two key models, the *Generator* and the *Discriminator*, GANs leverage a feedback loop between both models to refine their ability to generate relevant images. The Discriminator is trained to recognize images from a set of data. Properly trained, this model is able to distinguish between a real example, taken out of the dataset, from a “fake” image, foreign to the

dataset. The Generator, however, is trained to create images resembling images from the same dataset. As the Generator creates images, the Discriminator provides him with some feedback about the quality of its output. In response, the Generator adapts, to produce even more realistic images. Through this feedback loop, a GAN slowly builds up its ability to create relevant synthetic images, factoring in phenomena found among observed data.

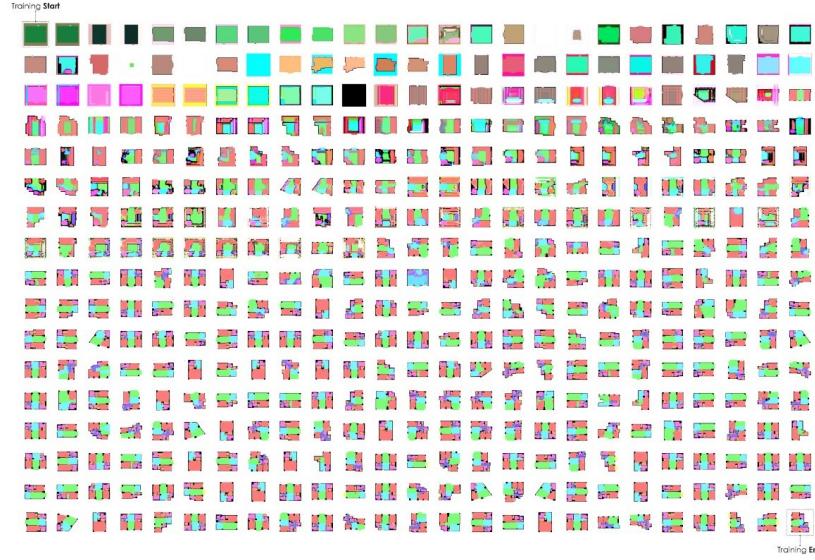


Generative Adversarial Neural Network's Architecture | Image Source

Representation & Learning

If GANs represent a tremendous opportunity for us, knowing what to show them is crucial. We have here the opportunity to let the model learn directly from floor plan images. By formatting images, we can control the type of information that the model will learn. As an example, just showing our model the shape of a parcel and associated building footprint will yield a model able to create typical building footprints given a parcel's shape. To ensure the quality of the outputs, we will use our own architectural "*sense*" to curate the content of our training sets: a model will only be as good as the data we give him, as architects.

Here below, we illustrate a typical training sequence: this sequence, realized over the course of a day and half of training, displays how one of our GAN-models progressively learns how to layout rooms and fenestration for a housing unit.



Training Sequence | Source: Author

Although the initial attempts are imprecise and confuse, after 250 iterations the machine builds for itself some form of intuition.

Precedents

If GANs' application to architectural design is still in its infancy, a handful of precedents inspired our work and drove our intuition. Hao Zheng's and Weixin Huang offered at the ACADIA conference in 2018 a first publication, demonstrating the potential of GAN for floorplans recognition and furniture layout generation. Using patches of color, their model would draw the infill of rooms, based on the room program, and its openings position. The same year, Nathan Peters in his thesis at the Harvard GSD, proposed to use GANs (pix2pix) to tackle program repartition in single-family modular homes, based on the house footprint.

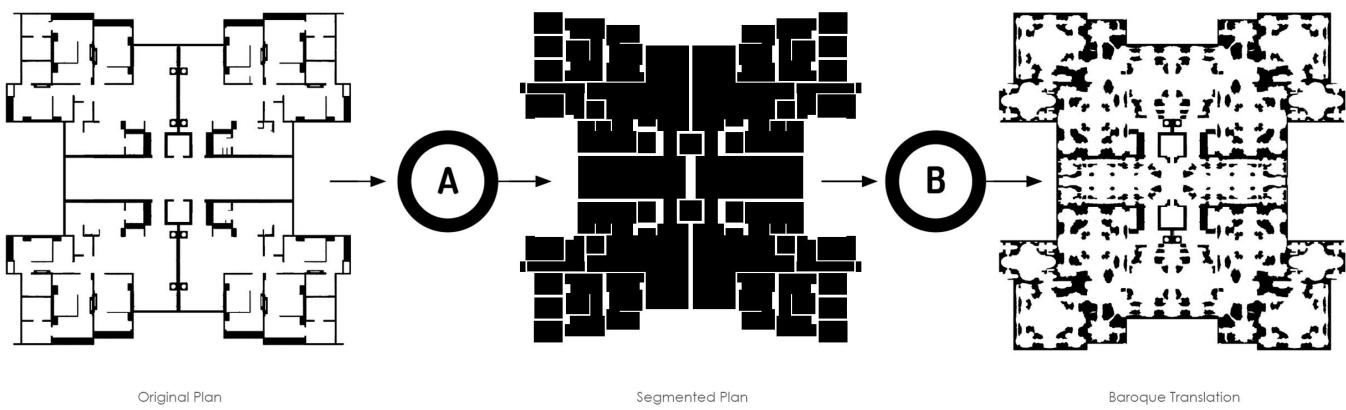
Regarding GANs as design assistants, Hao Zheng's (*Drawing with Bots: Human-computer Collaborative Drawing Experiments*, 2018) and Nono Martinez' thesis at the GSD (2017) inspired our investigations. Both authors tackled the idea of a loop between the machine and the designer to refine the very notion of "design process".

Our work expands on these precedents and offers to nest 3 models (footprint, program, and furnishing) to create a full "generation stack" while improving results quality at each step. By automating multi-units processing, our work then scales to entire buildings generation, and

masterplan layouts. We further offer an array of models dealing with style transfer. Finally, our contribution adds a rigorous framework to parse and classify resulting outputs, enabling users to “browse” consistently through generated options.

• • •

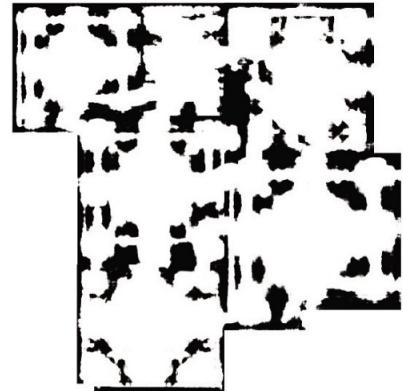
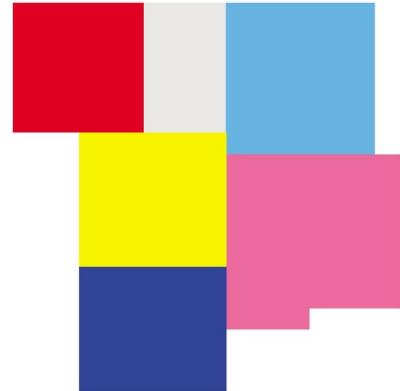
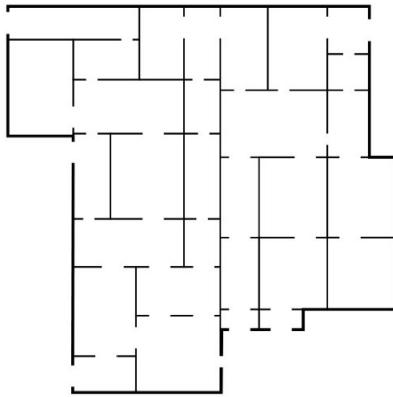
A. Style Transfer



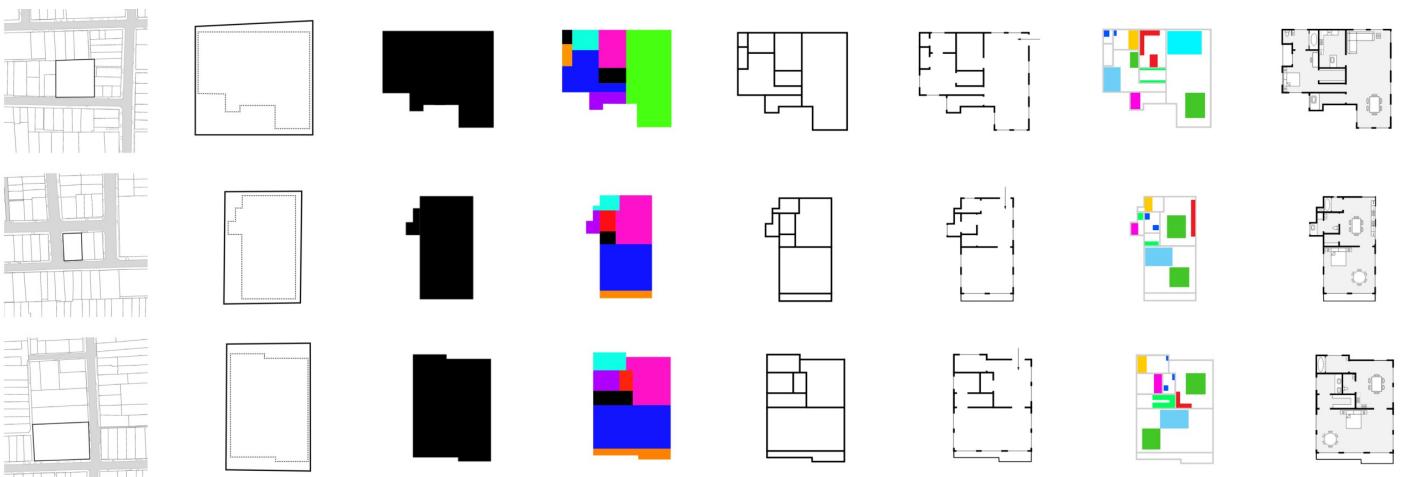
Modern-to-Baroque Floor Plan Translation | Source: Author

Within a floor plan, “Style” can be observed by studying the geometry and figure plane of its walls. Typical baroque churches will display bulky columns, with multiple round indents. A modern villa by Mies van der Rohe will show thin flat walls. This “*crenellation*” of the wall surface is a feature that a GAN can appreciate. By showing it pairs of images, with one image being a segmented version of a plan, and the other one the original wall structure, we can then build a certain amount of machine intuition with regard to architectural style.

This section shows the results of a model, trained to learn the Baroque style. We proceed then to a style transfer, where a given floor plan is *manually segmented (A)* and *dressed back with a new wall stylistic (B)*.



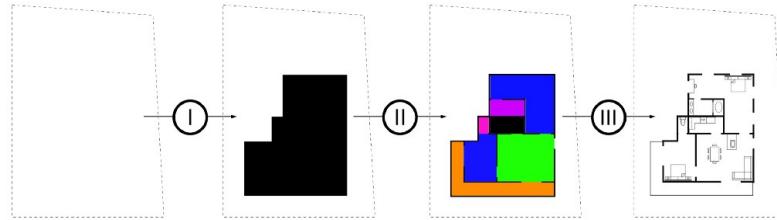
Style Transfer Results: Apartment Units **Modern-to-Baroque Style Transfer** | Source: Author



Layout Assistant, a Step by Step Pipeline | Source: Author

In this section, we offer a multi-step pipeline, integrating all the necessary steps to draw a floor plan. Jumping across scales, it emulates the process taken by an architect and tries to encapsulate each step into one specific model, trained to perform a given operation. From the **parcel to the building footprint (I)**, from the **footprint to a room split with walls & fenestration (II)**, from a **fenestrated floor plan to**

a furnished one (III): *each step has been carefully engineered, trained and tested.*



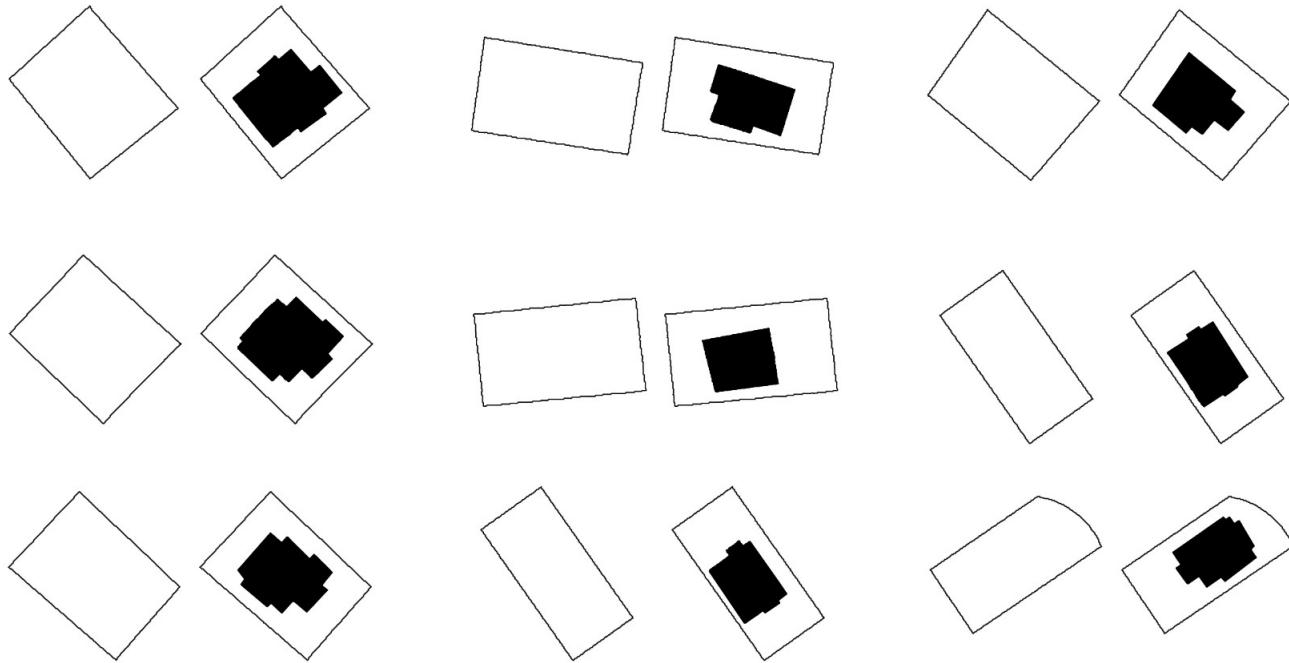
Generation Pipeline (Models I to III) | Source: Author

At the same time, by dividing the pipeline into discrete steps, the system allows for the user's intervention between each model. By selecting the output of a model, and editing it, before giving it to the next model, the user stays in control of the design process. Its input shapes the decisions made by the model, therefore achieving the human-machine interaction expected.

1. Footprint

The first step in our pipeline tackles the challenge of creating an appropriate building footprint for a given parcel geometry. To train this model, we used an extensive database of Boston's building footprints and were able to create an array of models, each tailored for a specific property type: *commercial, residential (house), residential (condo), industrial*, etc.

Each model is able for a given parcel, to create a set of relevant footprints, resembling in dimension and style the type it was trained for. 9 examples, using the residential (house) model are shown here below.

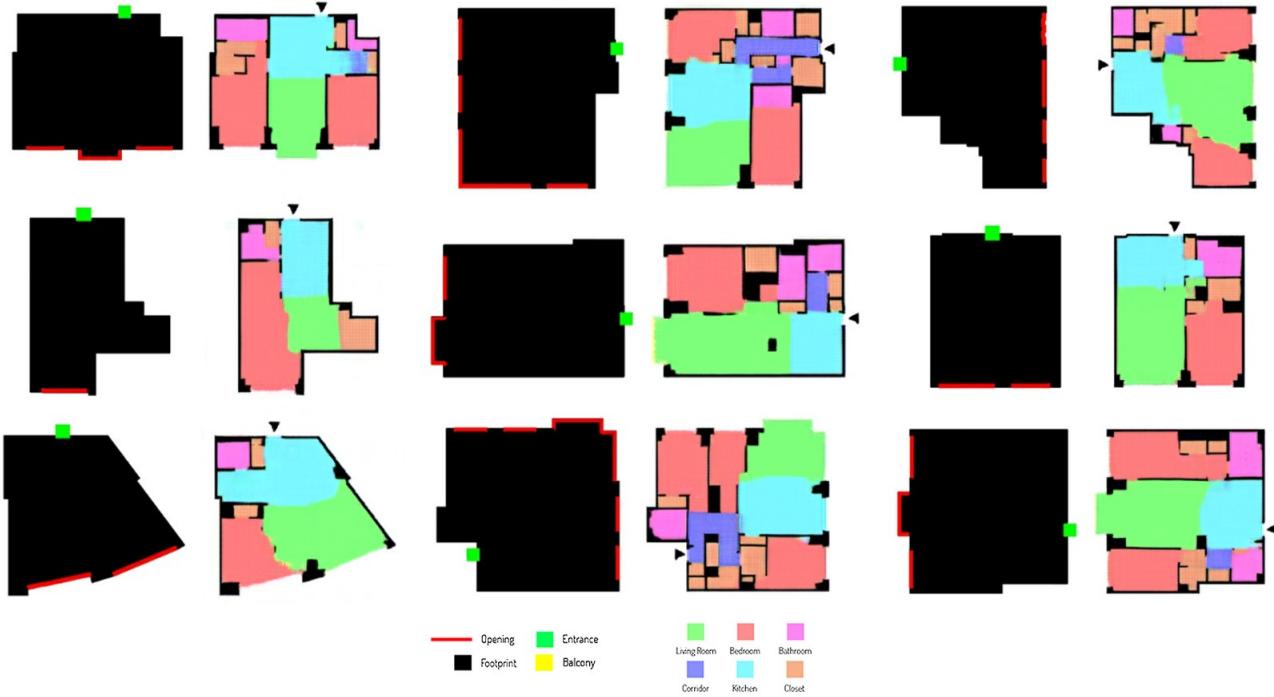


Results: Generated Footprints (housing) | Source: Author

2. Room Split & Fenestration

The layout of rooms across a building footprint is the natural next step. Being able to split a given floor plan, while respecting meaningful *adjacencies*, *typical room dimensions* and *proper fenestrations* is a challenging process, that GANs can tackle with surprising results.

Using a dataset of around 700+ annotated floor plans, we were able to train a broad array of models. Each is geared towards a specific room count and yields surprisingly relevant results once used on empty building footprints. We display here below some typical results. *More results are also available here.*



Results: Generated Program & Fenestration | Source: Author

3. Furnishing

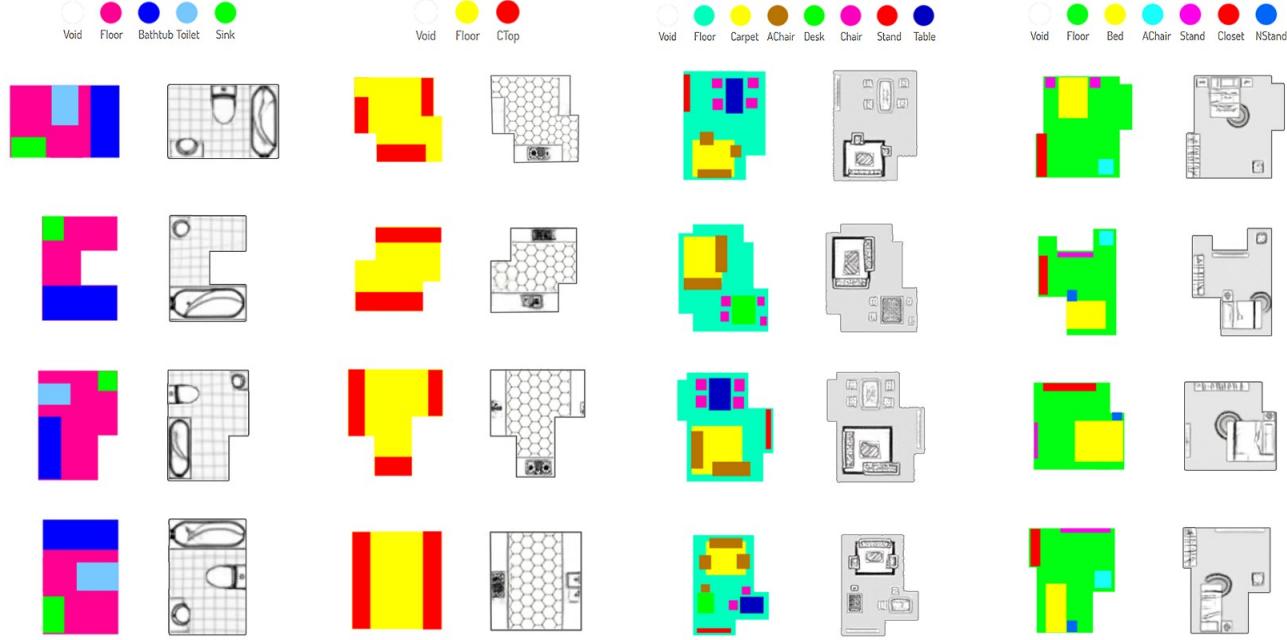
This last step brings the principle of generation to its most granular level: *the addition of furniture across space*. To that end, we trained at first a model to furnish the entire apartment all at once.

The network was able to learn, based on each room program, the relative disposition of furniture across space, and the dimensions of each element. The results are displayed here below.



Results: Furnished Units | Source: Author

If these results can give a rough idea of potential furniture layouts, the quality of the resulting drawings is still too fuzzy. To further refine the output quality, we have trained an array of additional models, for each room type (*living room, bedroom, kitchen, etc...*). Each model is only in charge of translating a color patch added onto the plan, into a properly drawn piece furniture. Furniture types are encoded using a color code. We display here below the results of each model.

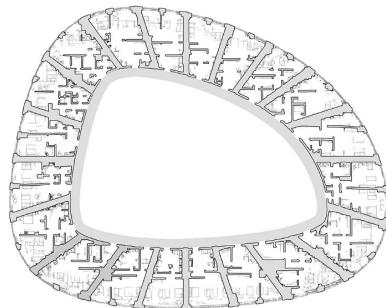
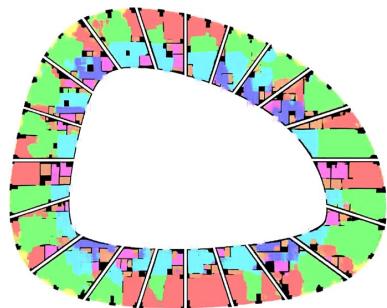
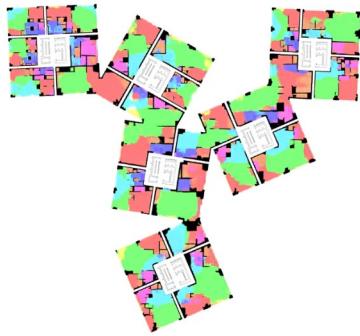
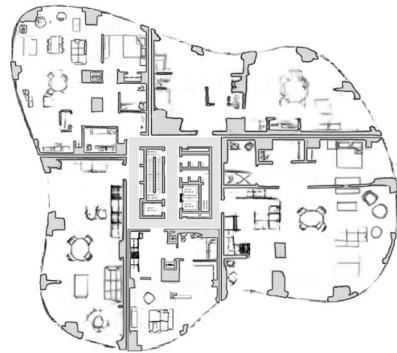
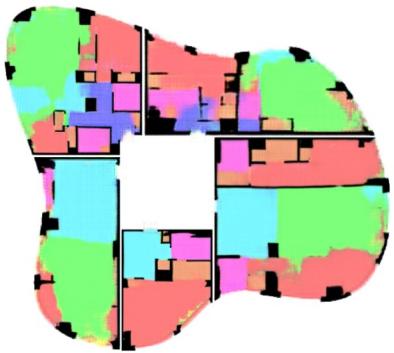


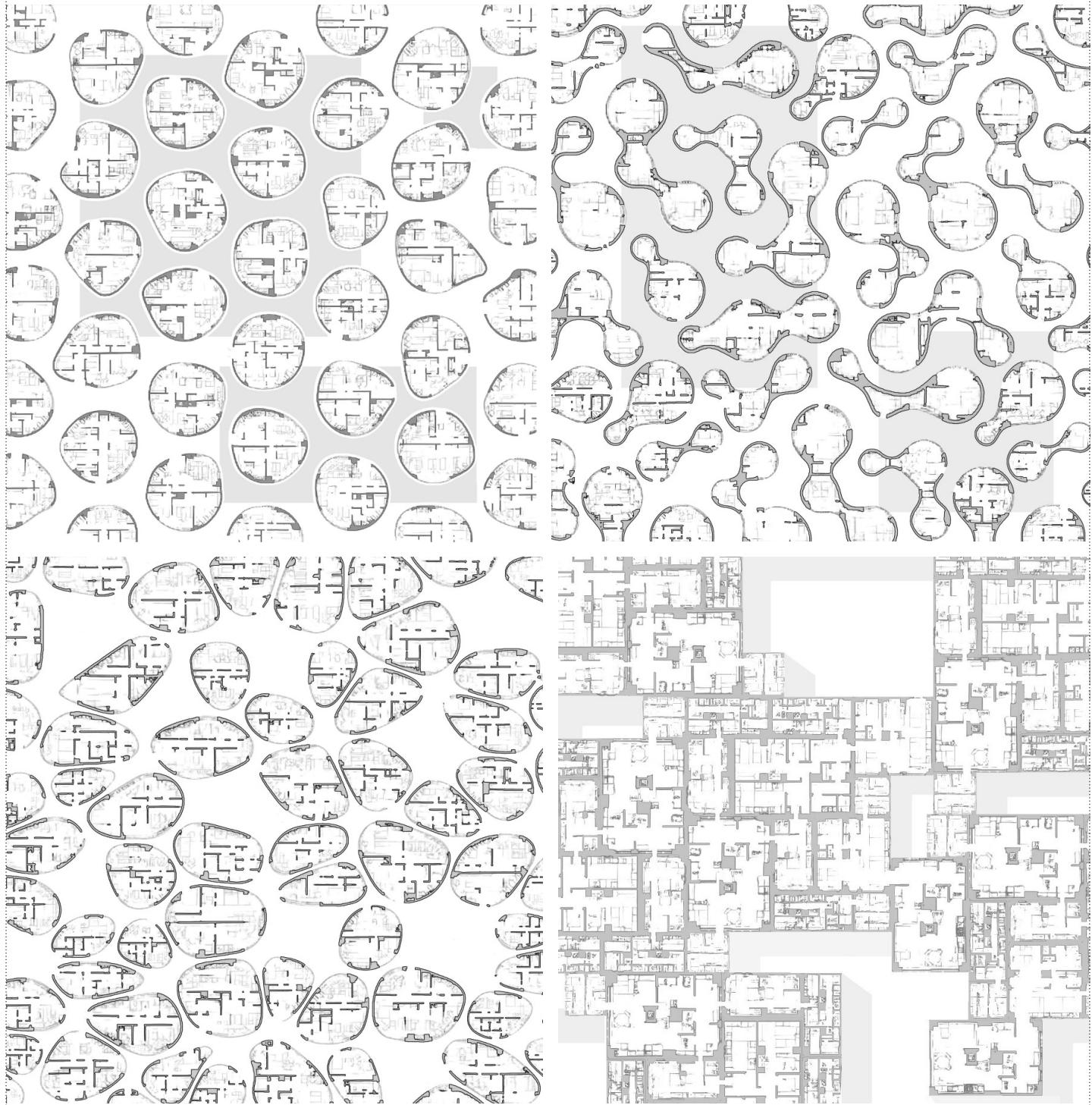
Results of Room Furnishing Models | Bathroom / Kitchen / Livingroom / Bedroom | Source: Author

4. Going Further

If generating standard apartments can be achieved using our technic, pushing the boundaries of our models is the natural next step. GANs can, in fact, offer quite remarkable flexibility to solve seemingly highly constrained problems. In the case of floor plans layout, as the footprint changes in dimension and shape, partitioning and furnishing the space by hand can be a challenging process. Our models prove here to be quite “smart” in their ability to adapt to changing constraints, as evidenced below.

Our ability to control the units' entrance door and windows position, coupled with the flexibility of our models allows us to tackle space planning at a larger scale, beyond the logic of a single unit. In the examples below we scale our technic to entire buildings.





Experimental GAN-generated Masterplans | Source: Author

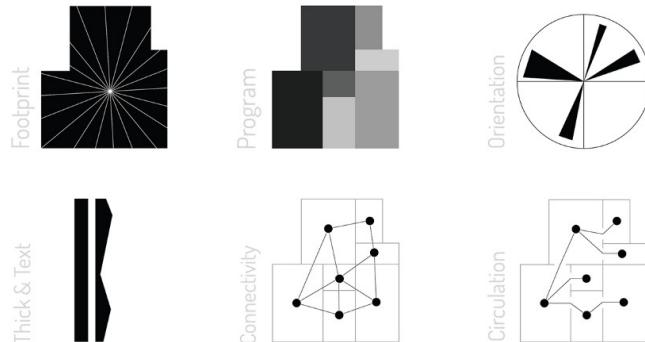
III. Qualify

"Failing to name things adds to the World's disarray"

Albert Camus

To balance out our ability to generate floor plans, **finding the proper framework to organize, sort & classify the wealth of generated design options is more than crucial.** The floor plans we offer will only be as good as our ability to navigate across our database of generated options. By borrowing concepts from Architecture, we hope to transform common architectural adjectives into quantifiable metrics.

To that end, we have isolated 6 key metrics, describing 6 essential aspects of floor plan design: **Footprint, Program, Orientation, Thickness & Texture, Connectivity and Circulation.**



6 Metrics | Source: Author

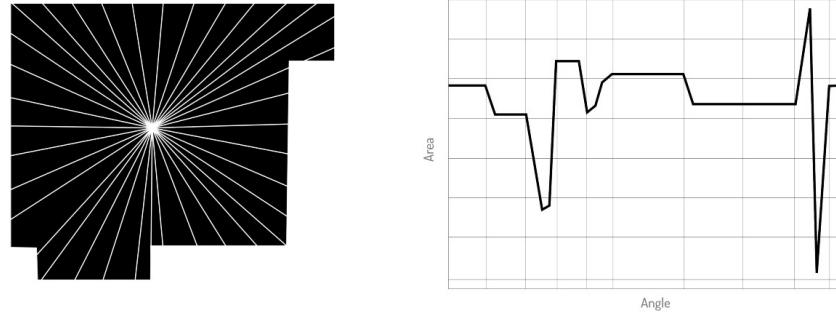
These metrics work together as a comprehensive framework, addressing both the stylistic and organizational dimensions of floor plans. Each one has been developed as an algorithm, and thoroughly tested.



A. Footprint

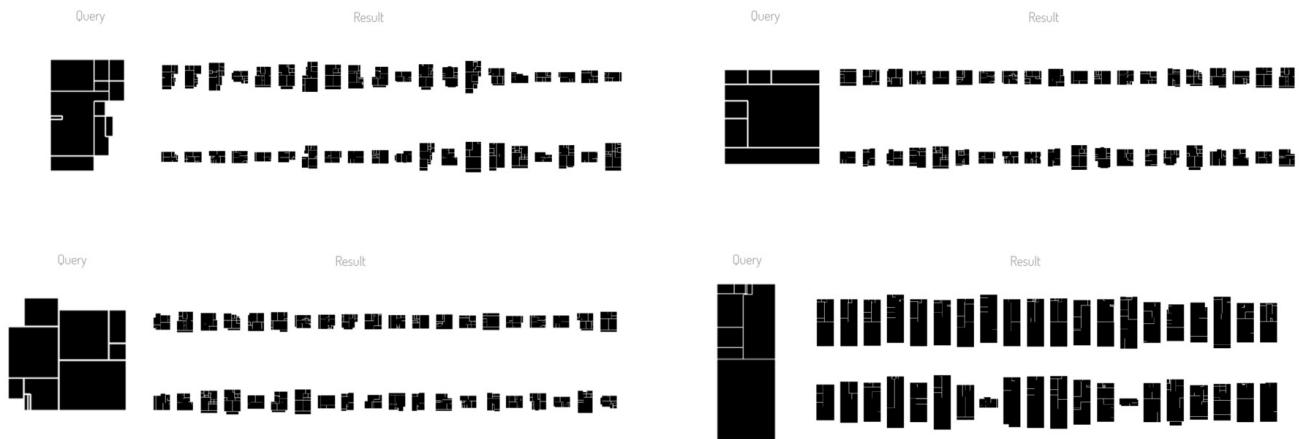
The shape of a building is the simplest and most intuitive proxy to qualify its style. The “Footprint” metric analyzes the shape of the floor plan perimeter and translates it into a histogram.

This descriptor, while encoding the shape of a building, can translate common adjectives—“thin”, “bulky”, “symmetrical”, etc—used by architects into numerical information, in order to communicate with a computer about building shapes.

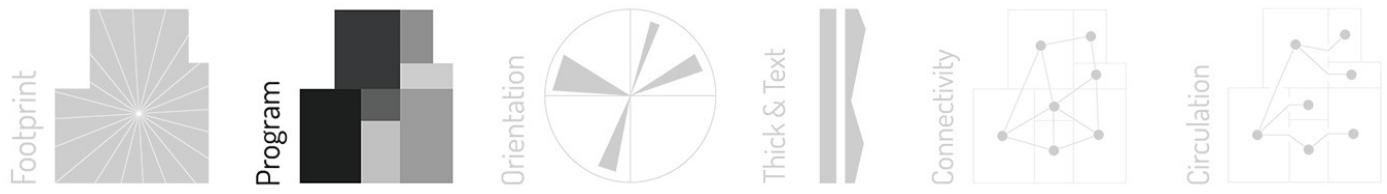


Footprint Polar Diagram | Source: Author

From a technical standpoint, this metric uses polar convexity to turn a given outline into a list of discrete values (vector) that can then be compared to other floor plans. We use a polar array of lines, stemming from the center of the plan, to extract the area of the plan captured by each slice of space obtained. This methodology has proven to yield satisfactory results, as shown in the queries here below. This technique can also be employed to qualify indoor spaces’ shape as well as building perimeter’s geometry.

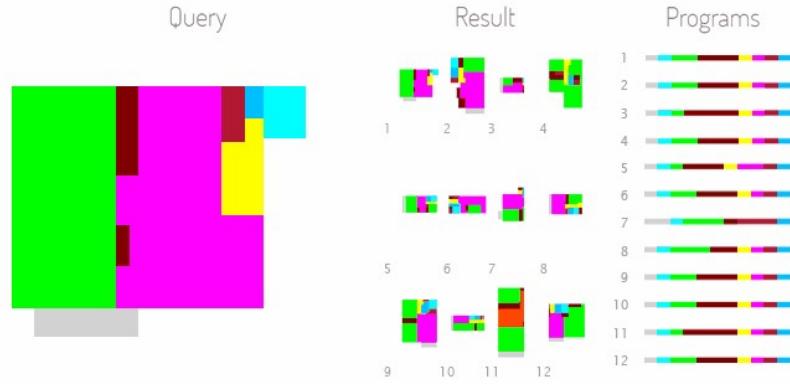


Typical Floor Plans Retrieval using the **Footprint** Metric (left: query, right: results) | Source: Author



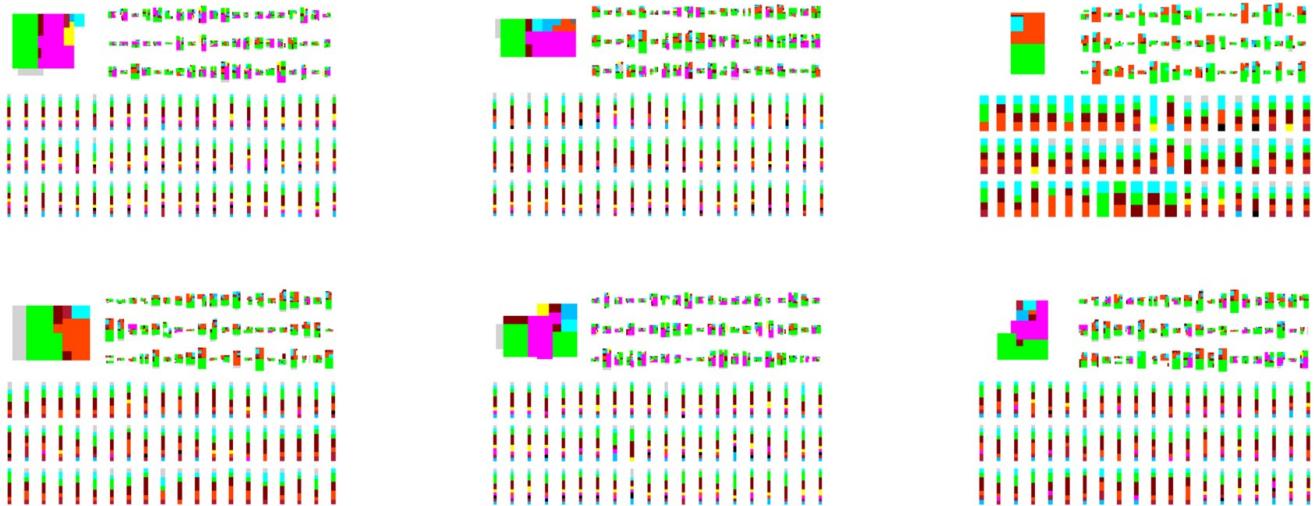
B. Program

The **Program** of a building, or in other words, the type of rooms it contains, is a major driver of its internal organization. Capturing this reality is central to our approach. To describe the “mix” of rooms we represent through a color code the list of rooms contained in any given floor plan. This colored band becomes the proxy, to describe the program. It acts as a template, aggregating both the quantity and the programmatic quality of the rooms within the floor plan. It is an intuitive visual description for humans which can translate into a reliable encoding technic for machines.

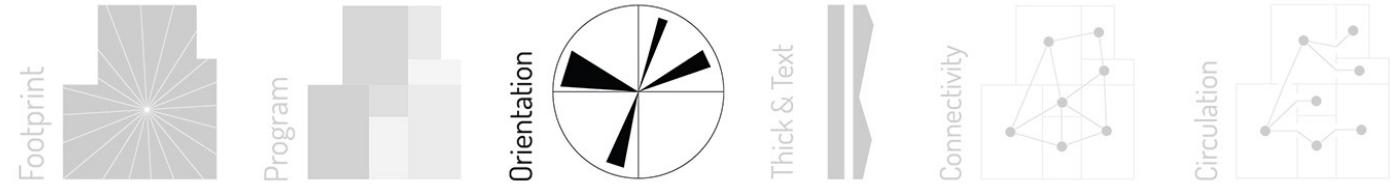


Typical Floor Plans Retrieval using the **Program** Metric (left query, right results) | Source: Author

From a technical standpoint, by using this colored band, we can compute the programmatic similarities and dissimilarities between any given pair of floor plans. To visualize results, each plan is reported as both a colored floor plan and a one-dimensional color vector of its program.

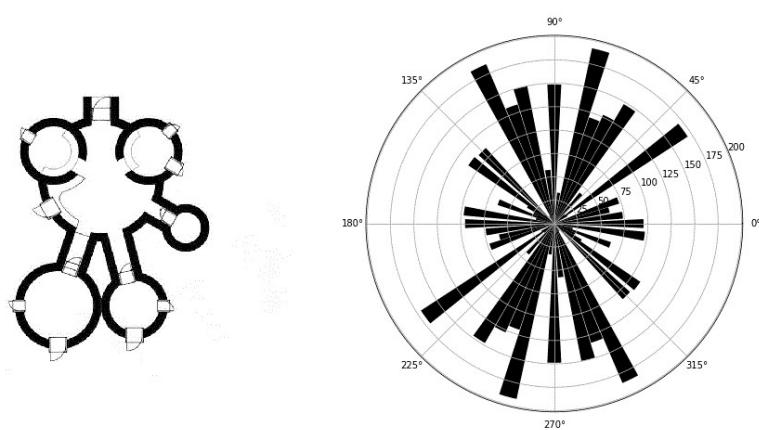


Typical Floor Plans Retrieval using the **Program** Metric (left: query, right: results, bottom: results' program) | Source: Author



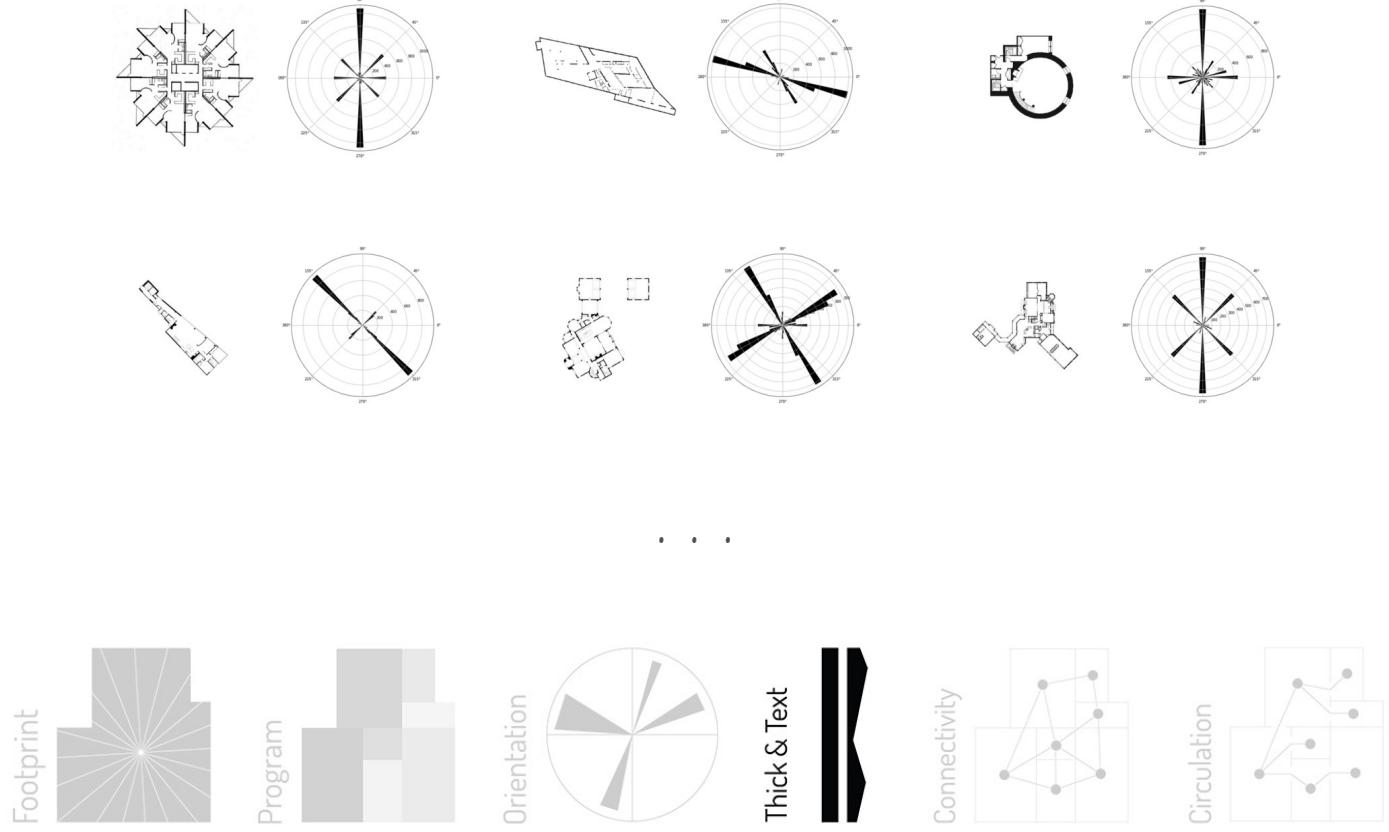
C. Orientation

The **Orientation** of walls in a plan is a valuable source of information. It can describe both the enclosure (how secluded spaces are due to the presence of walls) and the style of a plan. In fact, using this metric, we can easily differentiate a modern house-pavilion from a gothic cathedral, simply by extracting the histogram of the walls' orientation.



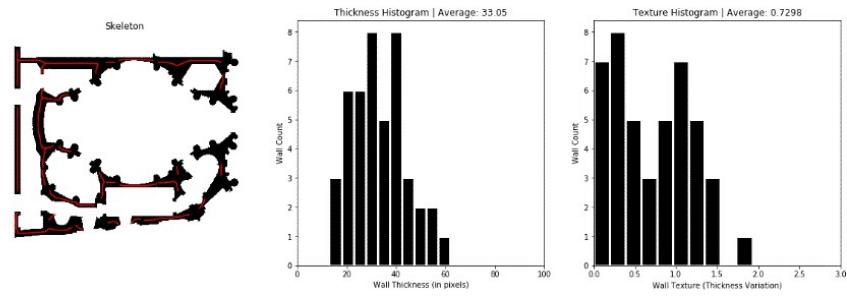
Orientation Diagrams | Source: Author

From a technical standpoint, orientation extracts the walls of a given floor plan and sums their length along each direction of space, from 0 to the 360 degrees. The resulting list of values is an assessment of the overall orientation of the plan. It can be averaged to get a single descriptor or used as a vector to compare across plans.



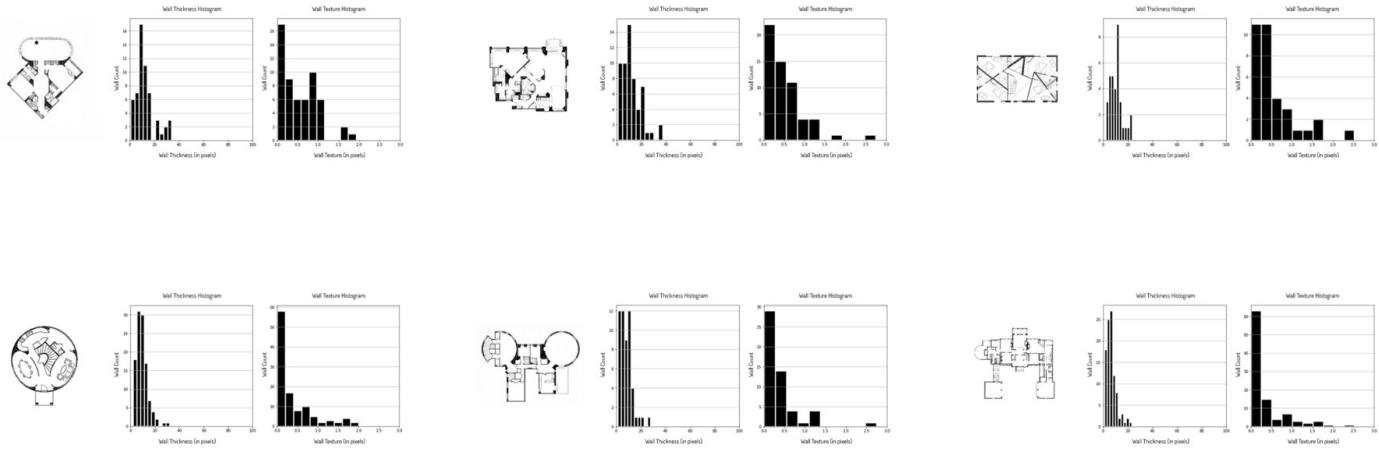
D. Thickness & Texture

The **Thickness & Texture** metric qualifies the “fat” of a plan: its wall thickness and the variation of this thickness. The *Thickness* of walls across a plan, as well as the geometry of the wall surface—*Texture*—can vary drastically from one style to another. A Beaux-Arts hall would display columns and indented thick walls when a villa from Mies van der Rohe would display thin rectilinear walls, which our metric would grasp easily (see image here below).



Thickness & Texture Diagrams | Source: Author

From a technical standpoint, this metric isolates all the walls of a given plan and outputs a histogram of wall thicknesses. At the same time, the algorithm computes the variation of the thickness, to better describe the wall texture (ie. flat walls versus mullions).

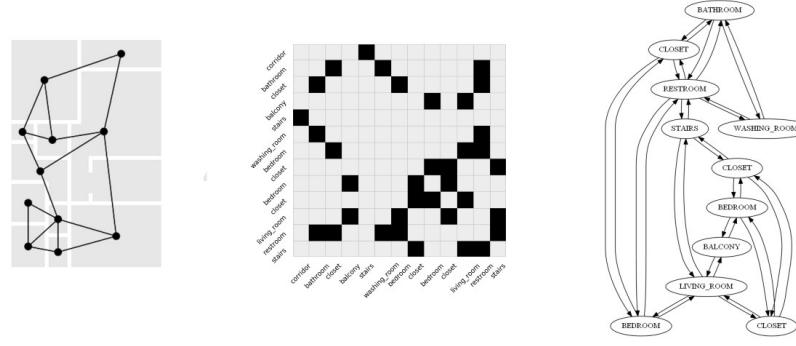


Typical Floor Plans Analysis using the **Thickness & Texture Metric** (left: plan, right: resulting diagrams) | Source: Author



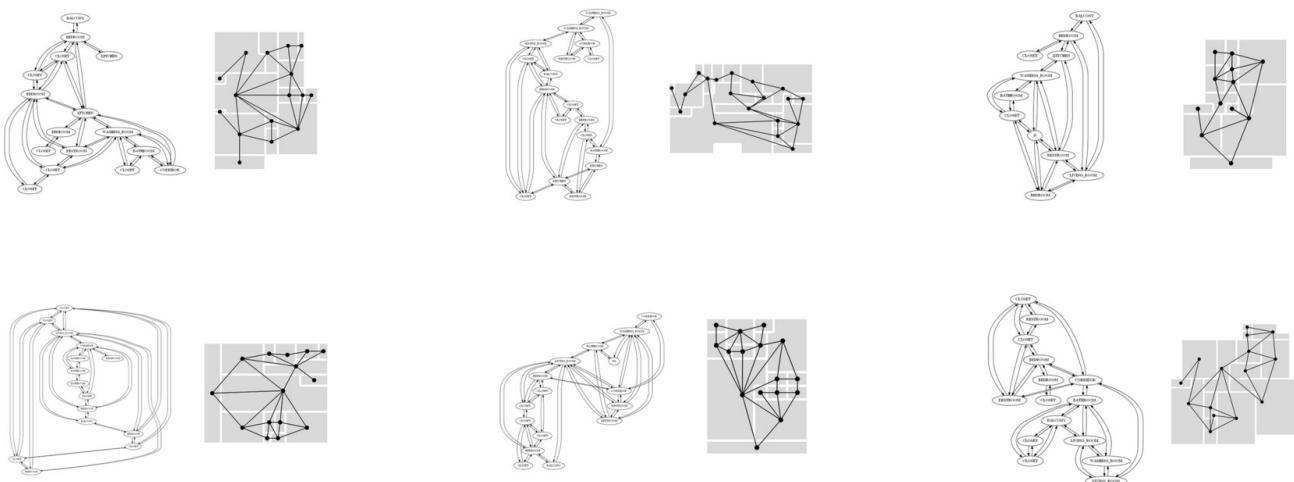
E. Connectivity

The **Connectivity** metric tackles the question of room adjacencies. The proximity of rooms to one another is a key dimension of a floor plan. Moreover, their connection through doors and corridors defines the existence of connections between them. **Connectivity** investigates the quantity and quality of such connections, by treating them as a standard graph.



Connectivity Diagram & Adjacency Matrix | Source: Author

From a technical standpoint, by using the fenestration from a plan, we can deduce the graph of existing relationships among rooms. The Connectivity metric then builds an adjacency matrix, reporting these connections. A graph representation is finally generated. Using this graph, we can compare floor plans, taking into account the similarity of connections among rooms.



Typical Floor Plans Analysis using the **Connectivity** Metric (left: connectivity graph, right: plan adjacencies) | Source: Author



Footprint



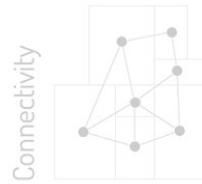
Program



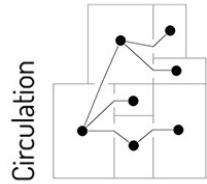
Orientation



Thick & Text



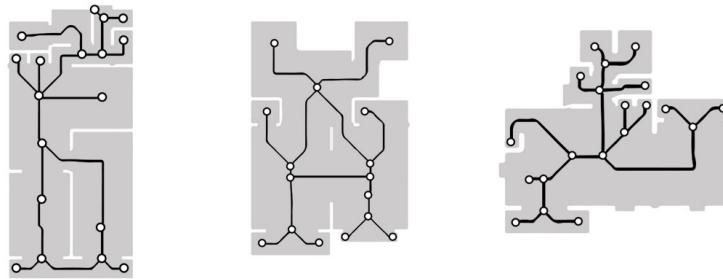
Connectivity



Circulation

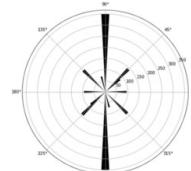
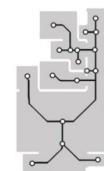
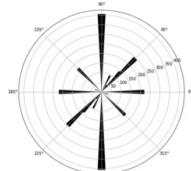
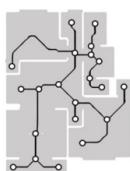
F. Circulation

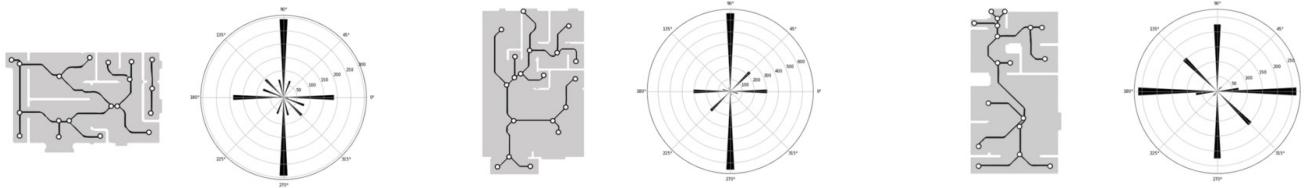
The **Circulation** in floor plans captures how people move across it. By extracting a skeleton of the circulation, or in other words, a wireframe of the circulatory network, we can both quantify and qualify people's movement across a floor plan.



Circulation Diagram | Source: Author

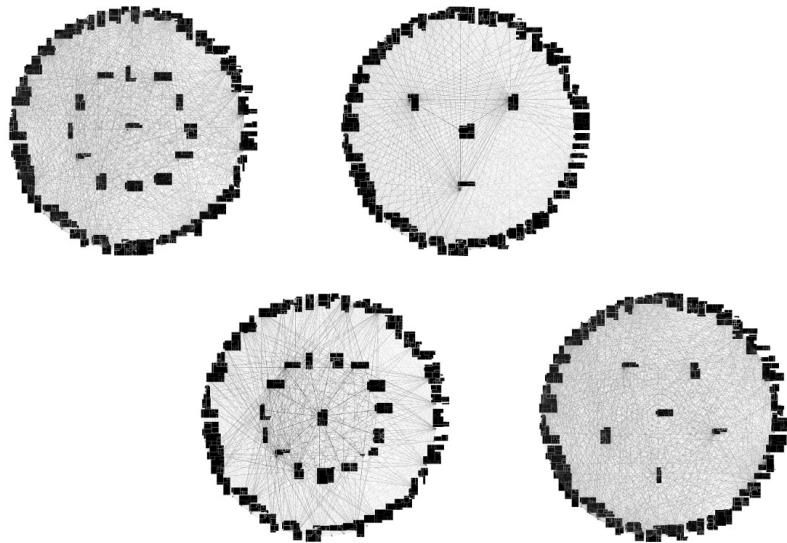
From a technical standpoint, Circulation extracts the skeleton of circulations of a given floor plan and sums its length along each direction of space, from 0 to 360 degrees. The resulting histogram is an assessment of the circulatory network geometry and can be used to be compared against other floor plans' circulation.





Typical Floor Plans Analysis using the **Circulation Metric** (left: circulation graph, right: diagram) | Source: Author

IV. Mapping & Browsing

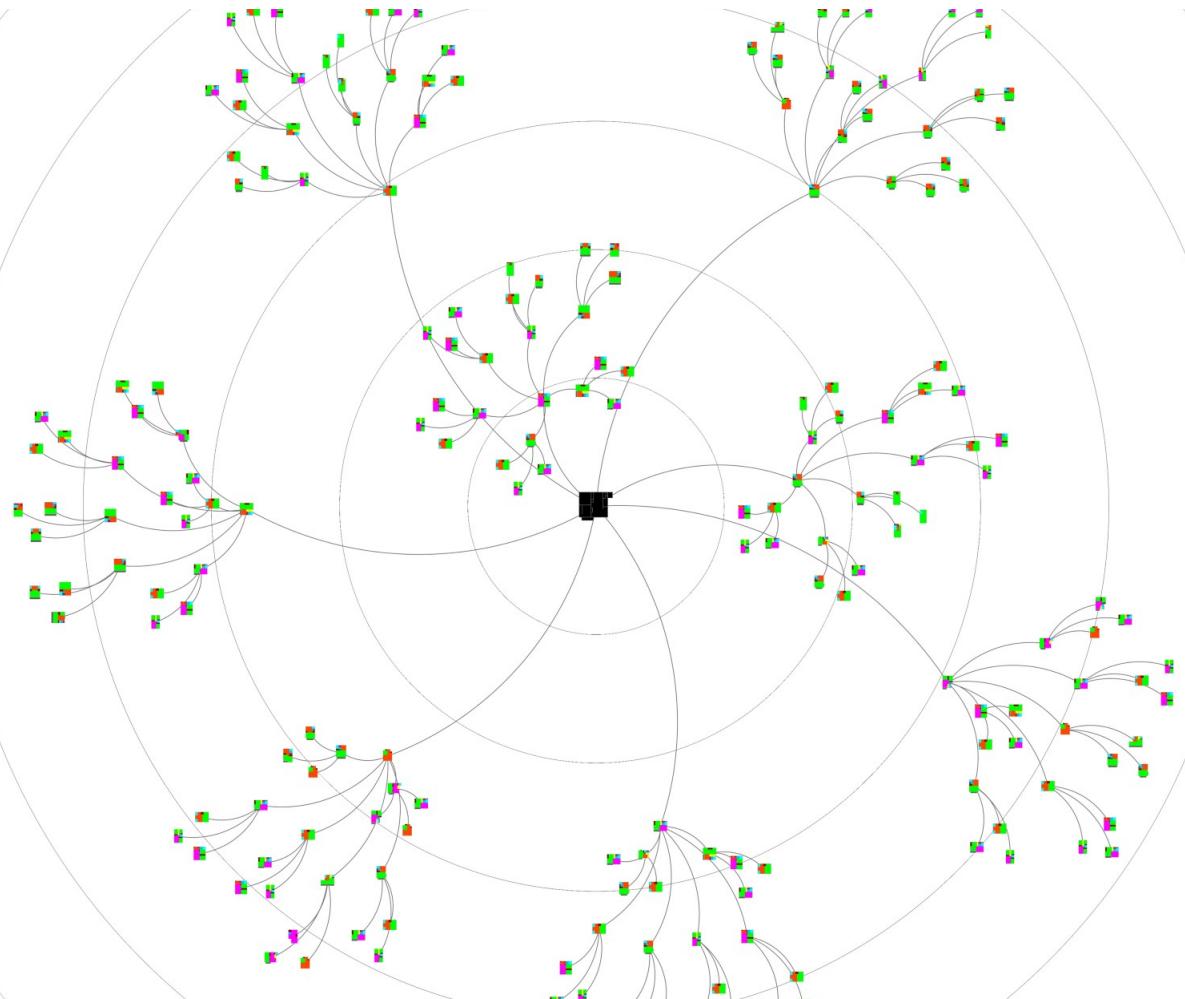


Similarity graphs, Comparing One Plan to Many | Source: Author

Looking back at our GAN-models, each one actually outputs multiple options at each step of our generation pipeline. The designer is then invited to “pick” a preferred option, modify it if needed, before actioning the next step. Browsing through the generated options however can be frustrating, and time-consuming. To that end, **the set of metrics defined in the “Qualify” chapter can demonstrate their full potential here and complement our generation pipeline.** By using them as filters, the user can narrow down the range of options and find in a matter of seconds the relevant option for its design. This

duality of **Generation-Filtering** is where the value of our work gets all the more evidenced: we provide here a complete framework, leveraging AI while staying within reach of a standard user.

Once filtered according to a given criterion (*Footprint, Program, Orientation, Thickness & Texture, Connectivity or Circulation*), we provide the user with a tree-like representation of her/his choice. At the center is a selected option, and around it, its nearest neighbors classified according to a user-selected criterion. The user can then narrow down the search and find its ideal design option, or select another option within the tree, to recompute the graph.



Floor Plans Similarity Trees | Source: Author

V. Conclusion

AI will soon massively empower architects in their day to day practice. As such potential is about to be demonstrated, our work participates to the proof of concept while our framework offers a springboard for discussion, *inviting architects to start engaging with AI, and data scientists to consider Architecture as a field of investigation.* However, today, our manifesto could be summarized in four major points.

Conceptually first, our belief is that a statistical approach to design conception shapes AI's potential for Architecture. Its less-deterministic and more-holistic character is undoubtedly a chance for our field. Rather than using machines to optimize a set of variables, relying on them to extract significant qualities and mimicking them all along the design process is a paradigm shift.

Second, we are directionally convinced that our ability to design the right pipeline will condition AI's success as a new architectural toolset. Our choice for the "Grey Boxing" approach, as introduced by Prof. Andrew Witt in Log, will likely secure the best potential results. This method contrasts with the "black box" approach, that only allows users to input information upfront, and to get finished design options at the end of the process, without any control over the successive generation steps. To the contrary, by breaking out our pipeline into discrete steps, "Grey Boxing" permits the user to intervene all along the way. His tight control over the machine is his ultimate guarantee of the design process quality.

Third, we technically believe that the sequential nature of the application will facilitate its manageability and foster its development. The ability to intervene throughout the generating process is a fundamental dimension: as each step of the pipeline represents a distinct portion of architectural expertise, each model can be trained independently, opening the way to significant improvements and experimentation in the near future. Indeed, improving this entire pipeline end-to-end could be a long and cumbersome task, while amending it step by step remains a manageable process, within the reach of most architects and engineers in the industry.

Finally, we hope our framework will help address the endless breadth and complexity of the models to be trained and those used in any generation pipeline. Tackling *parcels-footprint-room split-etc.*, as we do is one possible approach among, we believe, a large set of options. To encapsulate the necessary steps of space planning, the key is more the principle than the method. And with the growing availability of architectural data, we encourage further work and open-minded experimentation.

Far from thinking about AI as the new dogma in Architecture, we conceive this field as a new challenge, full of potential, and promises. We see here the possibility for rich results, that will complement our practice and address some blind spots of our discipline.

... .

Digital Resources



Thesis Website

Main Website of Thesis Project

<http://stanislaschaillou.com/expliquer>



Footprint Metric | Notebook

Algorithm & Applications of Footprint Metric

<https://github.com/StanislasChaillou/OpenPlan/blob/master/Footprint.ipynb>



Github Repository | Github

Main Code Repository of Thesis Project

<https://github.com/StanislasChaillou/OpenPlan>



Connectivity Metric | Notebook

Algorithm & Applications of Connectivity Metric

<https://github.com/StanislasChaillou/OpenPlan/blob/master/Connectivity.ipynb>



Connectivity Tree | Visualization

Similarity Tree computed according to Connectivity Metric

<http://stanislaschaillou.com/thesis/tree/graph>



Orientation Metric | Notebook

Algorithm & Applications of Orientation Metric

<https://github.com/StanislasChaillou/OpenPlan/blob/master/Orientation.ipynb>



Program Tree | Visualization

Similarity Tree computed according to Program Metric

<http://stanislaschaillou.com/thesis/tree/program>



Program Metric | Notebook

Algorithm & Applications of Program Metric

<https://github.com/StanislasChaillou/OpenPlan/blob/master/Program.ipynb>



Footprint Tree | Visualization

Similarity Tree computed according to Footprint Metric

<http://stanislaschaillou.com/thesis/tree/convexity>



Thickness & Texture Metric | Notebook

Algorithm & Applications of Thickness & Texture Metric

https://github.com/StanislasChaillou/OpenPlan/blob/master/Thickness_Texture.ipynb



Mapping Similarity | Notebook

Maps and Visualizations of floor plans similarities

https://github.com/StanislasChaillou/OpenPlan/blob/master/Mapping_Similarity.ipynb



Circulation Metric | Notebook

Algorithm & Applications of Circulation Metric

<https://github.com/StanislasChaillou/OpenPlan/blob/master/Circulation.ipynb>

• • •

Bibliography

Digital Architecture Beyond Computers, Roberto Botazzi,
Bloomsbury

Data-Driven Design & Construction, Randy Deutsch, Wiley

Architectural Intelligence, How Designers and Architects Created the Digital Landscape, Molly Wright Steenson, MIT Press

Architectural Google, Beyond the Grid—Architecture & Information Technology pp. 226–229, Ludger Hovestadt, Birkhauser

Algorithmic Complexity: Out of Nowhere, Complexity, Design Strategy & World View pp. 75–86, Andrea Gleiniger & Georg Vrachliotis, Birkhauser

Code & Machine, Code, Between Operation & Narration pp. 41–53,
Andrea Gleiniger & Georg Vrachliotis, Birkhauser

Gropius' Question or On Revealing And Concealing Code in Architecture And Art, Code, Between Operation & Narration pp. 75–89, Andrea Gleiniger & Georg Vrachliotis, Birkhauser

Soft Architecture Machines, Nicholas Negroponte, MIT Press.

The Architecture Machine, Nicholas Negroponte, MIT Press.

A Pattern Language, Notes on the Synthesis of Form,
Christopher Alexander, link

Cartogramic Metamorphologies; or Enter the RoweBot,
Andrew Witt, Log #36

Grey Boxing, Andrew Witt, Log #43

Suggestive Drawing Among Human and Artificial Intelligences, Nono Martinez, Harvard GSD Thesis, 2016

Enabling Alternative Architectures: Collaborative Frameworks for Participatory Design, Nathan Peters, Harvard GSD Thesis, 2017 | link

Architectural Drawings Recognition and Generation through Machine Learning, Hao Zheng (University of Pennsylvania), Weixin Huang (Tsinghua University), ACADIA 2018 [Paper]

DANIEL: A Deep Architecture for Automatic Analysis and Retrieval of Building Floor Plans, Divya Sharma, Nitin Gupta, Chiranjay Chattpadhyay, Sameep Mehta, 2017, IBM Research, IIT Jodhpur

Automatic Room Detection and Room Labeling from Architectural Floor Plans, Sheraz Ahmed, Marcus Liwicki, Markus Weber, Andreas Dengel, 2012, University of Kaiserslautern

Automatic Interpretation of Floor Plans Using Spatial Indexing, Hanan Samet, Aya Soffer, 1994, University of Maryland

Parsing Floor Plan Images, Samuel Dodge, Jiu Xu, Bjorn Stenger, 2016, Arizona State University, Rakuten Institute of Technology

Project Discover: An Application of Generative Design for Architectural Space Planning, Danil Nagy, Damon Lau, John

Locke, Jim Stoddart, Lorenzo Villaggi, Ray Wang, Dale Zhao and David Benjamin, 2016, The Living, Autodesk Studio

Raster-to-Vector: Revisiting Floor Plan Transformation, Chen Liu, Jiajun Wu, Pushmeet Kohli, Yasutaka Furukawa, 2017, Washington University, Deep Mind, MIT

Relational Models for Visual Understanding of Graphical Documents. Application to Architectural Drawings, Lluís-Pere de las Heras, 2014, Universitat Autònoma de Barcelona

Shape matching and modeling using skeletal context, Jun Xie, Pheng-Ann Heng, Mubarak Shah, 2007, University of Central Florida, Chinese University of Hong Kong

Statistical segmentation and structural recognition for floor plan interpretation, Lluís-Pere de las Heras, Sheraz Ahmed, Marcus Liwicki, Ernest Valveny, Gemma Sánchez, 2013, Computer Vision Center, Barcelona, Spain

Unsupervised and Notation-Independent Wall Segmentation in Floor Plans Using a Combination of Statistical and Structural Strategies, Lluís-Pere de las Heras, Ernest Valveny, and Gemma Sanchez, 2014, Computer Vision Center, Barcelona, Spain

Path Planning in Support of Smart Mobility Applications using Generative Adversarial Networks, Mohammadi, Mehdi, Ala Al-Fuqaha, and Jun-Seok Oh., 2018

Automatic Real-Time Generation of Floor Plans Based on Squarified Treemaps Algorithm, Fernando Marson and Soraia Raupp Musse, 2010, PUCRS

Procedural Modeling of Buildings, Pascal Muller, Peter Wonka, Simon Haegler, Andreas Ulmer, Luc Van Gool, 2015, ETH Zurich, Arizona State University

Generative Design for Architectural Space Planning, Lorenzo Villaggi, and Danil Nagy, 2017, Autodesk Research